AD-A115 498 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOO--ETC F/G 9/2
ANALYSIS OF THE OPTIMUM RECEIVER DESIGN PROBLEM USING INTERACTI--ETC(U)
DEC 81 M R MAZZUCCHI
UNCLASSIFIED AFIT/GE/EE/81D-39 NL

1 OF 3
AD
A115498

UNITED STATES AIR FORCE

AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GE/EE/81D-39

ANALYSIS OF THE OPTIMUM RECEIVER DESIGN PROBLEM USING INTERACTIVE COMPUTER GRAPHICS

THESIS

AFIT/GE/EE/81D-39 Michael R. Mazzucchi
Cpt USA

AFIT/GE/EE/81D-39

ANALYSIS OF THE OPTIMUM RECEIVER DESIGN PROBLEM USING INTERACTIVE COMPUTER GRAPHICS

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

by
Michael R. Mazzucchi, B.S.
Captain USA
Graduate Electrical Engineering
December 1981

Approved for public release; distribution unlimited.

Preface

The purpose of this study was to design a tool to assist engineering students learning signal detection theory and its application to communication receiver design. It was my own experience that essential insight is gained from working through several variations of a single problem and working several different such problems. This, however, requires much time and it is hoped this program will allow users to obtain this understanding with considerably less expenditure of time.

It is with sincere gratitude I acknowledge Major Kenneth Castor. He first introduced and schooled me in signal detection and estimation theory and it was his original proposal which resulted in this project. I am deeply indebted to him for his constant enthusiasm, his abundant recommendations, and enlightening discussions throughout our association.

Special thanks is due Professor Charles Richard who always stopped and took the time to listen to and help resolve the numerous programming problems.

Finally, I want to express my appreciation to my wife, Linda, not only for her encouragement and understanding all along the way, but also for the hours spent proofreading and for her many helpful suggestions.

Michael R. Mazzucchi

Contents

Page

List of Figures

List of Tables

Abstract

The purpose of this project was to design a tutorial aid for the study of signal detection theory and its application to communication receiver design. An interactive computer program was developed to solve problems concerning the detection of amplitude and/or phase shift keyed signals in the presence of additive white gaussian noise. The probability of error criterion was used to compare and optimize signal set parameters.

The user may input from 2 to 33 two-dimensional signal vectors ranging in amplitude from 10^{-6} to 10^{4} units, specify signal probabilities, and system noise energy. The computed system and signal statistics include signal energy, signal-to-noise ratio, Union Bound on and integrated values of probability of error, noise power spectral density, and center of gravity. Graphical displays provide signal set with coordinates and decision region boundaries. Modifications to signal set may be performed via translation, rotation, or scaling, and deletion or addition of signals.

The programming language used was FORTRAN 77 with graphical capability provided thru the Tektronix PLOT-10 graphics package. The program (less graphical capability) may be executed from any interactive terminal supported by the FORTRAN 77 compiler and the International Mathematical

& Statistical Libraries (IMSL) routines MDNOR and MDBNOR. For graphical displays, use of Tektronix terminals model 4014, 4012, or 4010 is required.

ANALYSIS OF THE OPTIMUM RECEIVER DESIGN PROBLEM USING INTERACTIVE COMPUTER GRAPHICS

I. Introduction

Today the world is spanned by a web of electrical circuits that permits near instantaneous communication over vast distances. The tools required to engineer this communication system are primarily decision and estimation theory. Some sort of "message" is generated at a source which results in an "observation" at a receiver. The message and observation are stochastically related and the objective is to determine a rule which forms a "best guess" of the message based on the observation. Of primary interest in communication theory then, is a method of distinguishing as accurately as possible a transmitted message by appropriate signal processing at the transmitter and deprocessing at the receiver.

Even though the foundations of detection and estimation theory are relatively recent (initial work by Wiener and Rice was done between 30 and 35 years ago), it is built on solid yet complex ground. It is this groundwork that all communication engineers must appreciate and understand. While the mathematics are not particularly difficult, the experience of countless graduate student engineers has

shown that the application and basic concepts of detection theory are. In his preface to Detection, Estimation, and Modulation, Part I, (Ref 7:ix), Harry L. VanTrees states:

> Throughout the course and book we emphasize the development of an ability to work problems. Only by working a fair number of them is it possible to appreciate the significance and generality of the results.

Because the solutions to these problems are lengthy and generally consist of numerous tedious calculations, many aspects of the solutions go unnoticed or unrecognized. Additionally, the problems most often must be restricted to the symmetric or standardized cases. Tools to facilitate working these problems and hence assist in the comprehension of detection theory are needed.

Problem

This effort is directed toward the creation of a tutorial tool to assist in the explanation and utilization of signal detection theory. The ultimate goal is the development of an interactive computer program for the analysis of communication receiver design problems. The program is intended to relieve the student of the tedium of problem solving thus allowing him to investigate and discover how altering the signal and channel parameters affect the probability of correctly receiving a transmitted message. In order for the student to better visualize the problem and and its solutions, a graphical display capability is to be provided.

Scope

The signal set chosen for this study is a combination of amplitude shift keyed (ASK) and/or phase shift keyed (PSK) signals. The program user will be able to input the following parameters:

1. Number of signals to be analyzed
2. Vector description of each signal
3. Probability of each signal
4. Power spectral density of system noise

After analysis, the following signal and system statistics will be provided as output:

1. Individual signal energies
2. Total energy
3. Center of gravity of signals
4. Display of signal set and decision regions
5. Conditional probability of error
6. Total probability of error

Additionally, the user will be able to modify the signal set by:

1. Translation and/or rotation of signal set
2. Altering original signal set by deleting, adding, and/or moving signals

Assumptions and Limitations

The theoretical complexity and nature of the problem demand that several assumptions be made.

1. The actual decision boundaries of any particular signal set depend on the a priori probabilities of the signals, the signals themselves, and the specifications of the channel transitional probabilities. The case in which the channel disturbs the signal vector by adding to it a random

noise vector will be the sole channel considered. The reasonable assumption that noise and signal components are statistically independent is used.

2. In order to simplify the decision function, the probability density of the noise component must be specified. In this project, the noise is considered to be white gaussian.

3. In formulating the optimum receiver design, it is assumed that the signals being transmitted and their probabilities are known at the receiver.

Approach and Presentation

In order to accomplish this task, three essential requirements were identified. The first was a good understanding of detection theory and the second was an adequate understanding of computer programming and computer graphics. Melding these two initial requirements, a program was produced which culminated in the third and final requirement. This last requirement then is the necessity for verifying that the program works as desired.

We begin in Chapter II by summarizing the theory of detection needed for the design of the program. Initially, we classify the detection problem and further delineate the assumptions used. Discussions of decision rules, the probability of error criterion, and the general gaussian problem provide the required fundamentals. A more in-depth study is

given in the following sections on the analysis of multiple signals and the application of the Union Bound.

With the basic concepts assumed, we move to Chapter III, "Concept and Design of the Software." Modular, top-down structured design was used throughout the program. The main program basically consists of eleven calls to subroutines or modules, which operate on the inputted data. Twelve other subroutines are user initiated for problem analysis, variation, and output display. As nearly as possible during the design phase, as each subroutine was completed, it was integrated into the main and validation tests run. The chapter begins with an overview of the program operation and then discusses the function and design of the individual subroutines.

In order to verify the program, several approaches could have been pursued. As Chapter IV explains, the procedure used here is the comparison of solutions of several typical problems. First the hand calculated result is explained and then the program generated solution presented and compared.

The concluding chapter contains a discussion of these results and areas for possible improvement or extension of capability.

II. Theory of Detection

Introduction

The reception of signals in noise presents problems of significant importance in the theory of communications, since noise to varying degrees always obscures the desired signal or message. Because the observation period during which the signal may be recovered is necessarily limited and because of the inherently statistical character of signal and interference, information is lost and recovery incomplete. Of course, reception of signals under such conditions can usually be carried out in a variety of ways, but very few of these possess optimum properties. The reception problem may then be described as the task of finding "best", or optimum, systems in order to remove the deleterious effects of the accompanying noise.

In communication theory, the reception problem is separated into two distinct fields. When the number of possible signals is finite, the problem is called a "decision" or "detection problem." If the number of signals is uncountably infinite, the problem is referred to as an "estimation problem."

Furthermore, reception is distinguished by the fact that the receiver has only a limited knowledge of the input signal (necessarily so, if any information is to be conveyed) and little or no control over it. In other words, the

judgement required about the input must be a statistical inference. This suggests the application of statistical decision theory for the design of optimum reception systems. Of primary importance is the criterion of excellence by which the performance of a reception system can be rated and with respect to which the optimization can be carried out. Once the criterion is selected, the optimum system is in principle determined. As mentioned in the introduction, many "optimum" criteria have been developed and the bulk of this chapter will be devoted to establishing the criterion upon which this project is based.

In order to attack a problem by statistical decision theory, we must have certain information available beforehand. We must have, for example, the statistics of the noise and if possible the statistics of the signals. The less we assume known concerning these, the more difficult is the solution in general, and the more general the solution. To justify the selection of the particular criterion used and to provide the reader with necessary terminology and background, a discussion of the theoretical approach to the problem will be presented.

Classifying the Detection Problem

The problem of the detection of a signal in noise is equivalent to one which, in statistical terminology, is called the problem of "testing hypothesis." Here, the hypothesis that the noise alone is present is to be tested,

on the basis of some received data, against the hypothesis (or hypotheses) that a signal (or one of several possible signals) is present.

Detection problems can be classified in a number of ways: by the number of possible signals which need to be distinguished, by the nature of the hypothesis or decision rule, by the nature of the data and their processing, and by the characteristics of the signal and noise statistics. Each of these classifications is discussed below and where appropriate, the assumptions used throughout this presentation will be addressed.

1. Number of Signals to be Distinguished. The number of signals to be distinguished is equal to the number of hypotheses to be tested. In "binary detection," one can make but two decisions, corresponding to the two hypotheses, while a "multiple alternative detection system" makes more than two decisions. The class of all possible (desired) system inputs is called the "signal class or set" and is conveniently represented as an abstract space in which each point corresponds to an individual signal. The set of possible signals is known as the signal space which is denoted by S. Thus, we can write:

$$S = \{s_1, s_2, \ldots., s_k\} \quad \text{or}$$

$$S = \{s_1(t), s_2(t), \ldots., s_k(t)\}$$

where $k \in \{2, 3, \ldots., 33\}$ for this project.

2. <u>The Nature of the Hypotheses or Decision Rules</u>. A signal is a desired system input. Noise is an undesired system input. The noise is considered to enter the system independently of the signal and to affect each observation according to the method whereby the two are combined. The combination method utilized here is addition, i.e., the observed signal has been perturbed by additive noise only. Thus, if the noise is denoted by n [or n(t)], we model the system as shown below.

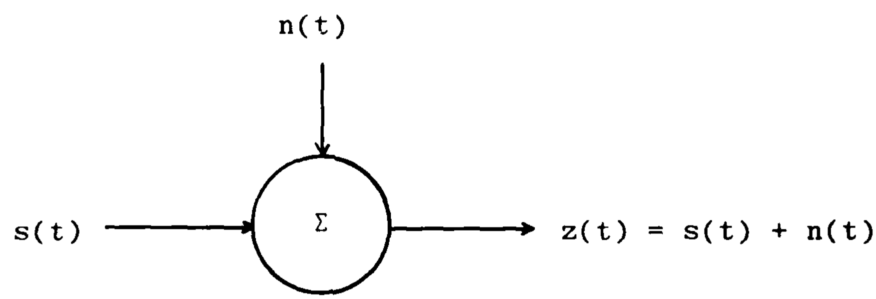


Fig 1. Communication System Model

3. <u>The Nature of the Data and Their Processing</u>. The observation made on the mixture of signal and noise during the observation period may consist of a discrete set of values (discrete or digital sampling) or may include a continuum of values throughout the interval (continuous or analog sampling). Whichever procedure is used is a characteristic of the problem. The set of possible observations

make up the observation space, Z, where

$$Z = \{z_1, z_2, \ldots., z_k\} \quad \text{or}$$

$$Z = \{z_1(t), z_2(t), \ldots., z_k(t)\}$$

Similarly, it is of interest whether or not the observation interval, i.e., the interval over which the reception system can store the data for analysis, is fixed or variable. In the latter case, one can consider "sequential" detection. In applications of decision theory, it turns out that the analysis divides conveniently at the choice between the sequential and the nonsequential. For the purpose of this project, the observations may include a continuum of values; however, only the nonsequential case will be considered.

4. The Signal and Noise Statistics. The nature of these quantities is clearly of central importance, as it is upon them that specific calculations of performance depend. The signal itself may be described in quite general terms involving both random and deterministic parameters. No restriction is placed on the signal other than that it have finite energy in an observation interval; it may be entirely random, partly random, or entirely deterministic. The description of the noise is necessarily statistical, and usually distinguished between noise belonging to "stationary" and "nonstationary" processes. As may be expected, it will

be assumed here that the signals being transmitted and their probabilities are known at the receiver. The noise will be considered to consist of independent, identically distributed, zero mean, gaussian random variables, each with identical variance, i.e., white gaussian noise. Additionally, the reasonable assumption that the noise and signal are statistically independent will be used.

The Decision Rule

As previously mentioned, the objective of the receiver is to take the observation and, using some predetermined rule, make a "best guess" at the transmitted message or signal. A brief discussion of this vital concept is necessary at this point, because it is this rule, known as the "decision rule," $d(z)$, that maps the observation space, Z, into the decision space, D, in some optimal manner. With the assumption that one of a finite set of signals is transmitted, the mapping $d: Z \to D$ is a partitioning of the observation space into "decision regions" corresponding to each element of Z. For example, for the binary signal set, the mapping is equivalent to dividing Z into two "decision regions," Z_1 and Z_2 such that $d(z) = d_1$ if z, the observation, is an element of region Z_1 and $d(z) = d_2$ if z is an element of region Z_2. The regions Z_1 and Z_2 must be disjoint in order that each point in Z will yield a unique decision. Additionally, Z_1 and Z_2 must cover Z,

($Z_1 \bigcup Z_2 = Z$), in order that each point in Z will have a decision associated with it.

As an example of a simple decision rule, the "Maximum-Likelihood Decision Criterion" will be developed. The basic concept of the maximum likelihood criterion is to select the decision corresponding to the message which is the most likely to have caused the observed signal. This technique requires knowledge of the conditional probability density functions of the observation given each of the possible messages, that is, $p(z|m)$. For the binary case, the criterion becomes: "Given an observation $z \varepsilon Z$, let $d(z) = d_1$ if it is more likely that m_1 generated z than that m_2 generated z" (Ref 3:22).

Mathematically speaking, the decision rule takes on the following form:

$$d(z) = \begin{cases} d_1 & \text{If} \quad p(z|m_1) > p(z|m_2) \\ d_2 & \text{If} \quad p(z|m_2) > p(z|m_1) \end{cases} \tag{1}$$

Therefore, given a particular observation z_o, one may compute $p(z_o|m_1)$ and $p(z_o|m_2)$ and apply the decision rule of Eq (1) to determine the transmitted signal. It should be noted that no assignment has been made when $p(z|m_1) = p(z|m_2)$. The values of z for which the conditional densities are equal may be arbitrarily assigned to either d_1 or d_2 since m_1 and m_2 are equally likely to have been the cause of the observed z.

Sometimes the application of a decision rule can be simplified by performing mathematical operations on the conditional densities. As previously noted, an equivalent method for representing the decision rule of Eq (1) is to define the decision regions Z_1 and Z_2 as:

$$Z_1 = \{z : p(z|m_1) > p(z|m_2)\}$$

$$Z_2 = \{z : p(z|m_2) > p(z|m_1)\} \tag{2}$$

If the "Likelihood Ratio" $\Lambda(z)$ is defined as

$$\Lambda(z) = \frac{p(z|m_2)}{p(z|m_1)} \tag{3}$$

then Z_1 and Z_2 may be defined as:

$$Z_1 = \{z : \Lambda(z) < 1\}$$

$$Z_2 = \{z : \Lambda(z) > 1\} \tag{4}$$

Using shorthand notation, the decision rule becomes:

$$\Lambda(z) \underset{d_1}{\overset{d_2}{\gtrless}} 1 \tag{5}$$

One may operate on the likelihood ratio expression as long as the unique ordering of $\Lambda(z)$ relative to unity is maintained. The natural logarithm is quite often a useful operator especially for gaussian problems.

The decision rule has thus become a simple method of assigning a message value to each received observation. It sets an easily implemented threshold which a receiver can use to decide which signal was transmitted.

Discussion of the Probability of Error Criterion

It was mentioned in the introduction that several different criteria have been developed to form different types of decision rules. Before presenting the particular criterion to be used in this project, it is necessary to introduce some definitions and notation. For ease of explanation, the discussion will consider the binary case initially, and appropriate references to the multiple decision case will be interjected.

In dealing with the binary decision problem, there are two types of errors that one can make. First, one may decide d_2 when m_1 is true, and second, one may decide d_1 when m_2 is true. Each of these errors has a probability associated with it which depends on the decision rule and conditional densities. The following notation will be employed:

$P\{d_2|m_1\}$ = Probability of making decision d_2 when m_1 is true

$P\{d_1|m_2\}$ = Probability of making decision d_1 when m_2 is true

In addition to two errors, there are also two correct decisions that one can make in the binary decision problem. One may decide d_1 when m_1 is true and one may decide d_2 when m_2 is true. Again, these correct decisions have associated probabilities represented by:

$P\{d_1|m_1\}$ = Probability of making decision d_1 when m_1 is true

$P\{d_2|m_2\}$ = Probability of making decision d_2 when m_2 is true

The probability $P\{d_2|m_1\}$ is sometimes referred to as the "false-alarm probability," or in terms of statistical decision theory, as the "level of significance." The probability $P\{d_2|m_2\}$ is sometimes referred to as the "detection probability" or as the "power of the test." Finally, $P\{d_1|m_2\}$ is referred to as the "miss probability."

This notation will be used throughout this project as it is simple and compact, yet complete. Where necessary, specific probability densities and/or subscripts will be added to prevent ambiguity.

The decision-rule to be used is known as the "Minimum Probability of Error," P_e, criterion which can be defined as:

$$
\begin{aligned}
P_e &= P\{\text{making an incorrect decision}\} \\
&= P\{\text{decide } d_2 \text{ when } m_1 \text{ is true or decide } d_1 \text{ when } m_2 \text{ is true}\} \\
&= P\{(d_2 \text{ and } m_1) \text{ or } (d_1 \text{ and } m_2)\}
\end{aligned}
$$

The probability of error criterion says to select the decision regions so as to minimize this total probability of error. Realizing that the messages m_1 and m_2 are mutually exclusive and using conditional probabilities, the P_e can be written:

$$P_e = P\{d_2|m_1\}\ P\{m_1\} + P\{d_1|m_2\}\ P\{m_2\} \tag{6}$$

Here, $P\{m_1\}$ and $P\{m_2\}$ are the a priori probabilities and represent the probability that message m_k will be the message selected for transmission. Since m_1 and m_2 thru m_k are mutually exclusive and exhaustive, it must be that

$$P\{m_1\} + P\{m_2\} + \dots + P\{m_k\} = 1$$

When $P\{m_1\} = P\{m_2\} = \dots = P\{m_k\}$, the signal set under study is referred to as the "equally likely message" case. In order to select the decision regions to minimize the P_e, one must be able to write P_e in terms of the decision regions. For the binary case, we begin with Z_2. Once Z_2 is specified, Z_1 is automatically described and one minimizes P_e by selecting only those observations, z, which should be in Z_2 as message m_2. For example,

$$P\{d_1|m_2\} = \int_{Z_1} p(z|m_2)dz \tag{7}$$

which says that the probability of error, i.e., deciding d_1 given m_2 can be found by integrating the conditional

probability $p(z|m_2)$ over the bounds or region of Z_1. Since $P\{d_1|m_2\} + P\{d_2|m_2\} = 1$, it follows that:

$$P\{d_1|m_2\} = 1 - \int_{Z_2} p(z|m_2)dz \tag{8}$$

Similarly, one can write

$$P\{d_2|m_1\} = \int_{Z_2} p(z|m_1)dz \tag{9}$$

It is easily shown (Ref 3:39), that in order to minimize the probability of error, Eq (3) repeated below is

$$\Lambda(z) = \frac{p(z|m_2)}{p(z|m_1)} \tag{10}$$

used to provide a mathematical description of the decision regions as

$$Z_2 = \{z : \Lambda(z) > \frac{P\{m_1\}}{P\{m_2\}} \}$$

$$Z_1 = \{z : \Lambda(z) < \frac{P\{m_1\}}{P\{m_2\}} \} \tag{11}$$

and correspondingly the decision rule,

$$\Lambda(z) \underset{d_1}{\overset{d_2}{\gtrless}} \frac{P\{m_1\}}{P\{m_2\}} \tag{12}$$

Here again, the decision rule consists of comparing the likelihood ratio to a threshold; however, the threshold is determined by the ratio of the a priori probabilities.

The probability-of-error decision rule has another interesting interpretation. Consider the general rule given above written as follows:

$$\frac{p(z|m_2)}{p(z|m_1)} \underset{d_1}{\overset{d_2}{\gtrless}} \frac{P\{m_1\}}{P\{m_2\}} \tag{13}$$

Multiplying both sides of this expression by $\frac{P\{m_2\}}{P\{m_1\}}$, one obtains

$$\frac{p(z|m_2)\ P\{m_2\}}{p(z|m_1)\ P\{m_1\}} \underset{d_1}{\overset{d_2}{\gtrless}} 1 \tag{14}$$

By using the mixed Bayes rule, this can be rewritten as

$$\frac{P(m_2|z)p(z)}{P(m_1|z)p(z)} = \frac{P(m_2|z)}{P(m_1|z)} \underset{d_1}{\overset{d_2}{\gtrless}} 1 \tag{15}$$

This result says to decide d_1 if $P(m_1|z) > P(m_2|z)$ and decide d_2 if $P(m_2|z) > P(m_1|z)$. In other words, one should select the decision corresponding to the message with the larger a posteriori probability, i.e., the probability of m_k given z. Hence, the probability-of-error criterion is identical to the maximum a posteriori (MAP) decision criterion stated as follows: *Given an observation, z, select* d_1 *if*

m_1 is more likely than m_2. The key feature of this decision situation is that $p(z|m)$ becomes a rule for making the decision, d, from a posteriori data alone, i.e., without knowledge of or dependence upon the particular message, m, that resulted in the observation z. The a priori knowledge of the signal set and the signal distribution is built into the optimum decision rule. *Thus, the decision rule becomes the mathematical embodiment of the physical system used to process the data and yield decisions.*

The Concept of the Sufficient Statistic

In order to broaden the class of binary decision problems to include those where the observation is more complicated than a single scalar, the maximum likelihood function $\Lambda(z)$ becomes a function of a vector rather than a scalar. That is

$$\Lambda(\underline{z}) = \frac{p(\underline{z}|m_2)}{p(\underline{z}|m_1)} \underset{d_1}{\overset{d_2}{\gtrless}} \lambda$$

where the value of λ is determined by the particular decision criterion. One must now consider more than one observation, but the probability-of-error criterion continues to be just a likelihood ratio test. All that needs to change is that the integrals over the decision regions now become I-fold integrals rather than simple one-dimensional integrals. Hence, regardless of the dimensionality of the vector $\underline{z}$, the

decision rule can be formulated as a threshold test on the likelihood ratio $\Lambda(\underline{z})$. Since the decision rule is a mapping from the observation space Z to the decision space D, any operator on $\underline{z}$ that produces the same mapping can be used as a decision rule. A "sufficient statistic" is defined as a function $\ell(z)$ such that any likelihood ratio decision rule d(z) can be written as a function of $\ell(z)$ (Ref 3:64). The concept of the sufficient statistic can be quite useful in simplifying decision-rule implementation and analyzing system performance. It must be noted that since the threshold in the likelihood-ratio test can take on any value, and since a sufficient statistic must be able to mirror this test, it must be possible to determine the value of the likelihood ratio from the sufficient statistic. In other words, it provides enough information about the observation to enable a decision to be made.

Discussion of the General Gaussian Problem

Several approaches could be taken to extend the development of detection theory. As mentioned earlier, there are numerous classifications of the detection problem, and in order to solve a particular problem, specific assumptions are generally made. In lieu of examining each of these cases individually, the general gaussian problem will be solved and then the assumptions used in this project and described earlier will be applied. In this manner, the

validity and generality of the assumptions can be highlighted. The only constraint will be that the conditional probability density function of the observation, $\underline{z}$, be gaussian.

The general form of the density function of an I^{th}-order gaussian vector, $\underline{z}$, with mean, $\underline{s}$, and variance matrix, $\underline{V}$, is

$$p(\underline{z}) = \frac{1}{(2\pi)^{I/2} (\det \underline{V})^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(\underline{z} - \underline{s})^T \underline{V}^{-1} (\underline{z} - \underline{s})\right] \quad (16)$$

where it is assumed that $\underline{V}$ is positive definite and symmetric (Ref 3:69). The components of $\underline{s}$ are the expected values of the components of $\underline{z}$:

$$E\{z_i\} = s_i \qquad \text{For } i = 1,2,\ldots,I$$

The elements of $\underline{V}$ are the covariances of z_i and z_j:

$$\begin{aligned} v_{ij} &= E\{(z_i - s_i)(z_j - s_j)\} \\ &= \operatorname{cov}(z_i, z_j) \qquad \text{For } i,j = 1,2,\ldots,I \end{aligned}$$

Since the random variables, z_i, are jointly gaussian, the conditional densities, $p(\underline{z}|m_1)$ and $p(\underline{z}|m_2)$ can be written:

$$p(\underline{z}|m_1) = \frac{1}{(2\pi)^{I/2} (\det \underline{V}_1)^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(\underline{z} - \underline{s}_1)^T \underline{V}_1^{-1} (\underline{z} - \underline{s}_1)\right] \quad (17)$$

and
$$p(\underline{z}|m_2) = \frac{1}{(2\pi)^{I/2}\,(\det \underline{V}_2)^{\frac{1}{2}}}$$

$$\exp\left[-\frac{1}{2}(\underline{z} - \underline{s}_2)^T\, V_2\, (\underline{z} - \underline{s}_2)\right] \tag{18}$$

The likelihood ratio is therefore

$$\Lambda(\underline{z}) = \frac{(\det \underline{V}_1)^{\frac{1}{2}} \exp\left[-\frac{1}{2}(\underline{z} - \underline{s}_2)^T\, \underline{V}_2^{-1}\, (\underline{z} - \underline{s}_2)\right]}{(\det \underline{V}_2)^{\frac{1}{2}} \exp\left[-\frac{1}{2}(\underline{z} - \underline{s}_1)^T\, \underline{V}_1^{-1}\, (\underline{z} - \underline{s}_1)\right]} \tag{19}$$

And as before, the likelihood ratio test is

$$\Lambda(\underline{z}) \underset{d_1}{\overset{d_2}{\gtrless}} \lambda \tag{20}$$

By appropriately assigning the value of λ, any threshold test can be performed. In order to show this, the logarithm of Eqs (19) and (20) is taken and the likelihood ratio test for the general gaussian problem becomes

$$(\underline{z} - \underline{s}_1)^T\, \underline{V}_1^{-1}\, (\underline{z} - \underline{s}_1)$$

$$- (\underline{z} - \underline{s}_2)^T\, \underline{V}_2^{-1}\, (\underline{z} - \underline{s}_2) \underset{d_1}{\overset{d_2}{\gtrless}} 2\ln\lambda + \ln \frac{\det \underline{V}_2}{\det \underline{V}_1} \tag{21}$$

In this most general form, Eq (21) is difficult to evaluate; hence, the necessity for making the various assumptions. Whenever the noise is independent of the

signal being sent, the assumption of equal covariance matrices is valid. That is,

$$\underline{V}_1 = \underline{V}_2 = \underline{V}$$

Equation (21) is thus simplified to

$$(\underline{z} - \underline{s}_1)^T \underline{V}^{-1} (\underline{z} - \underline{s}_1) - (\underline{z} - \underline{s}_2)^T \underline{V}^{-1} (\underline{z} - \underline{s}_2) \underset{d_1}{\overset{d_2}{\gtrless}} 2\ln\lambda \qquad (22)$$

In order to determine the sufficient statistic, the matrices are multiplied out and the common terms combined as follows:

We begin with the first term of Eq (22), which becomes

$$\begin{aligned}(\underline{z} - \underline{s}_1)^T \underline{V}^{-1} (\underline{z} - \underline{s}_1) &= (\underline{z}^T \underline{V}^{-1} - \underline{s}_1^T \underline{V}^{-1})(\underline{z} - \underline{s}_1) \\ &= \underline{z}^T \underline{V}^{-1} \underline{z} - \underline{s}_1^T \underline{V}^{-1} \underline{z} \\ &\quad - \underline{z}^T \underline{V}^{-1} \underline{s}_1 + \underline{s}_1^T \underline{V}^{-1} \underline{s}_1\end{aligned}$$

Similarly, for the second term

$$\begin{aligned}(\underline{z} - \underline{s}_2)^T \underline{V}^{-1} (\underline{z} - \underline{s}_2) &= \underline{z}^T \underline{V}^{-1} \underline{z} - \underline{s}_2^T \underline{V}^{-1}\underline{z} \\ &\quad - \underline{z}^T \underline{V}^{-1} \underline{s}_2 + \underline{s}_2^T \underline{V}^{-1} \underline{s}_1\end{aligned}$$

Substituting back into Eq (22), combining like terms, and

using matrix algebra to simplify, yields

$$2(\underline{s}_2 - \underline{s}_1)^T \underline{V}^{-1} \underline{z} \underset{d_1}{\overset{d_2}{\gtrless}} 2\ln\lambda + \underline{s}_2^T \underline{V}^{-1} \underline{s}_2 - \underline{s}_1^T \underline{V}^{-1} \underline{s}_1 \qquad (23)$$

Which ultimately provides the desired test,

$$\ell(\underline{z}) = (\underline{s}_2 - \underline{s}_1)^T \underline{V}^{-1} \underline{z} \underset{d_1}{\overset{d_2}{\gtrless}} \ln\lambda + \tfrac{1}{2}[\underline{s}_2^T \underline{V}^{-1} \underline{s}_2 - \underline{s}_1^T \underline{V}^{-1} \underline{s}_1] = \lambda' \qquad (24)$$

Now since $\ell(\underline{z})$ is a linear combination of jointly gaussian random variables, it is in fact gaussian, and determining its statistics requires knowledge of only its mean and variance. Taking the expected values and performing normal operations yields the relations below for the necessary statistics. The complete derivations of these quantities can be found in Appendix A. By defining $\Delta\underline{s}$ as

$$\Delta\underline{s} \triangleq \underline{s}_2 - \underline{s}_1$$

the variance of $\ell(\underline{z})$ can be shown to be

$$\mathrm{Var}\{\ell | m_k\} = \Delta\underline{s}^T \underline{V}^{-1} \Delta\underline{s} \qquad (25)$$

And the expected value of ℓ *given that* m_1 *is true becomes*

$$E\{\ell | m_1\} = \Delta\underline{s}^T \underline{V}^{-1} \underline{s}_1 \qquad (26)$$

Similarly, given that m_2 is true, gives

$$E\{\ell|m_2\} = \Delta\underline{s}^T \underline{V}^{-1} \underline{s}_2 \tag{27}$$

As stated previously, the probability of false alarm is given by:

$$P_{FA} \triangleq P\{d_2|m_1\} = \int_{\lambda'}^{\infty} p(\ell|m_1)\, d\ell \tag{9}$$

From the preceding results (e.g., Eq (17)), we know that $p(\ell|m_1)$ can be written

$$p(\ell|m_1) = \frac{1}{(2\pi)^{\frac{1}{2}} (\Delta\underline{s}^T \underline{V}^{-1} \Delta\underline{s})^{\frac{1}{2}}} \exp\left[-\frac{1}{2} \frac{(\ell - \Delta\underline{s}^T \underline{V}^{-1} \underline{s}_1)^2}{\Delta\underline{s}^T \underline{V}^{-1} \Delta\underline{s}}\right] \tag{28}$$

By substitution, we obtain:

$$P_{FA} = \int_{\lambda'}^{\infty} \frac{1}{(2\pi)^{\frac{1}{2}} (\Delta\underline{s}^T \underline{V}^{-1} \Delta\underline{s})^{\frac{1}{2}}} \exp\left[-\frac{1}{2} \frac{(\ell - \Delta\underline{s}^T V^{-1} \underline{s}_1)^2}{\Delta\underline{s}\, \underline{V}^{-1} \Delta\underline{s}}\right] d\ell \tag{29}$$

This expression can be simplified to become

$$P_{FA} = Q\left[\frac{\ln\lambda}{(\Delta\underline{s}^T \underline{V}^{-1} \Delta\underline{s})^{\frac{1}{2}}} + \frac{(\Delta\underline{s}^T V^{-1} \Delta\underline{s})^{\frac{1}{2}}}{2}\right] \tag{30}$$

where the function Q( ) is related to the error function and defined as in Ref (10:49).

As before, the derivation of this simplification is located in Appendix A.

If we now define

$$\delta^2 \triangleq \Delta\underline{s}^T \, V^{-1} \, \Delta\underline{s}$$

the probability of false alarm, P_{FA}, can be written

$$P_{FA} \triangleq P\{d_2 | m_1\} = Q\left(\frac{\delta}{2} + \frac{\ln\lambda}{\delta}\right) \qquad (31)$$

And the probability of detection, P_D, can be written

$$P_D \triangleq P\{d_1 | m_2\} = Q\left(\frac{\delta}{2} - \frac{\ln\lambda}{\delta}\right) \qquad (32)$$

If the variance matrix, $\underline{V}$, is a product of a scalar, v, and the identity matrix, $\underline{I}$, (i.e., independent noise components with different variances), the problem becomes similar to the one used in this project. That is, the problem models the addition of an independent gaussian noise of variance, v, to each component of s. The sufficient statistic becomes

$$\ell(\underline{z}) = \frac{1}{v} \Delta\underline{s}^T \, \underline{z} \qquad (33)$$

and the squared distance becomes

$$\delta^2 = \frac{\| \Delta\underline{s} \|^2}{v} \qquad (34)$$

The appearance of v in the demoninator of δ^2 indicates that the performance of a system would decrease as v increases. Hence, the system performance is determined by the ratio of $\|\Delta \underline{s}\|^2$ to v. If $\|\Delta \underline{s}\|^2$ is taken as a measure of signal energy, then δ^2 becomes a measure of the signal-to-noise ratio (Ref 3:72).

Equations (31), (32), (33), and (34) thus become the theoretical basis for the remainder of the project. Given the signal and noise statistics and energies, a sufficient statistic can be determined and compared against a given threshold to provide a solution to the decision rule. Additionally, these same values allow computation of the probabilities of error to be used as a basis for optimizing a particular solution.

Decision Rule for Multiple Signals

In order to extend these results to the multiple decision case, the following assumptions are made:

1. The message space, M, consists of K messages, i.e., $M = \{m_1, m_2, \ldots, m_k\}$.

2. The signal space, S, consists of K signals, i.e., $S = \{s_1, s_2, \ldots, s_k\}$, where there is a one to one mapping between M and S.

3. The decision space, D, consists of J elements, i.e., $D = \{d_1, d_2, \ldots, d_J\}$, where J is usually equal to K, but this is not a necessary condition.

4. The observation space, Z, will be partitioned into J regions such that $\bigcup_{J=1}^{J} Z_j = Z$, $Z_j \bigcap Z_k = \phi$ for all $j \neq k$, and if z is an element of Z_j, the decision, d, will be denoted by d_j.

The probability of error criterion can be considered a subset of the Bayes Decision Criterion. The Bayes criterion employs a systematic procedure of assigning a "cost" to each correct and incorrect decision and then minimizes the total average cost. Associated with each message, m_k, and decision, d_j, there is a unique cost, C_{jk}, which is defined as the cost of deciding d_j given the message was m_k. By utilizing this concept of "cost", the minimization of the probability of error can be achieved. The Bayes cost or risk becomes the expected value or average cost associated with a particular problem. Mathematically, this is

$$B = E\{C_{jk}\}$$

From the definition of expected value, this becomes

$$B = \sum_{j=1}^{J} \sum_{k=1}^{K} C_{jk} \; P\{d_j \, , \, m_k\}$$

$$= \sum_{j=1}^{J} \sum_{k=1}^{K} C_{jk} \; P\{d_j | m_k\} \; P\{m_k\} \tag{35}$$

By substitution of Eq (9) into Eq (35), the Bayes risk can be rewritten

$$B = \sum_{j=1}^{J} \sum_{k=1}^{K} C_{jk} P\{m_k\} \int_{Z_j} p(\underline{z}|m_k) \, d\underline{z} \tag{36}$$

Taking the summation over K inside the integral, we have

$$B = \sum_{j=1}^{J} \int_{Z_j} \sum_{k=1}^{K} C_{jk} P\{m_k\} p(\underline{z}|m_k) \, d\underline{z} \tag{37}$$

The Bayes criterion says to select the decision regions Z_j, $j = 1, 2, \ldots, J$ such that the average cost or risk is minimized. It can be shown (Ref 3:97,98) that B is minimized by selecting Z_j such that z is an element of Z_j if

$$\sum_{k=1}^{K} C_{jk} P\{m_k\} p(\underline{z}|m_k) < \sum_{k=1}^{K} C_{\ell k} P\{m_k\} p(\underline{z}|m_k) \tag{38}$$

for all $\ell \neq j$.

If we require that the number of messages be equal to the number of decisions and that there be a logical pairing of message m_i to decision d_i, several simplifications to Eq (38) can be made. Since we are concerned with the probability of error criterion, we will assume probability of error costs given by

$$C_{jk} = \begin{cases} 0 & \text{If} \quad j = k \\ 1 & \text{If} \quad j \neq k \end{cases}$$

Substituting this into Eq (38), we find that z will be an element of Z_j if

$$\sum_{\substack{k=1 \\ k \neq j}}^{K} P\{m_k\}\, p(\underline{z}|m_k) < \sum_{\substack{k=1 \\ k \neq \ell}}^{K} P\{m_k\}\, p(\underline{z}|m_k) \tag{39}$$

for all $\ell \neq k$. We note that the sums on both sides of Eq (39) are identical except that a different term is missing. If we add and subtract $P\{m_j\}\, p(\underline{z}|m_j)$ on the left hand side of Eq (39) and add and subtract $P\{m_\ell\}\, p(\underline{z}|m_\ell)$ on the right hand side, we have

$$\sum_{k=1}^{K} P\{m_k\}\, p(\underline{z}|m_k) - P\{m_j\}\, p(\underline{z}|m_j)$$

$$< \sum_{k=1}^{K} P\{m_k\}\, p(\underline{z}|m_k) - P\{m_\ell\}\, p(\underline{z}|m_\ell) \tag{40}$$

Cancelling the common term and changing signs yields the following definition of decision region Z_j:

$$P\{m_j\}\, p(\underline{z}|m_j) > P\{m_\ell\}\, p(\underline{z}|m_\ell) \tag{41}$$

for all $\ell \neq j$.

Equation (41) says that we should compute $P\{m_k\}\, p(\underline{z}|m_k)$ for $k = 1, 2, \ldots, K$ and then select the decision corresponding to the value of k for which $P\{m_k\}\, p(\underline{z}|m_k)$ is maximum. In other words, z is an element of Z_j if

$$P\{m_j\}\ p(\underline{z}|m_j) = \max_k P\{m_k\}\ p(\underline{z}|m_k) \tag{42}$$

Equation (42) thus becomes the general decision rule for the probability of error of multiple signals. By assuming that [illegible] corrupted by the addition of white gaussian noise, this result can be applied to the project under consideration.

For an additive-noise problem, the observation is just the sum of the signal vector and a noise vector

$$\underline{z} = \underline{s}_k + \underline{n} \tag{43}$$

Therefore, the conditional probability density function of $\underline{z}$, as before, is the density of $\underline{n}$ shifted to be centered at $\underline{s}_k$:

$$p(\underline{z}|m_k) = p_{\underline{n}}(\underline{z} - \underline{s}_k) \tag{44}$$

Now, if the noise is white and gaussian, the conditional density can be written as

$$p(\underline{z}|m_k) = (2\pi\sigma^2)^{-I/2} \exp - \frac{\| \underline{z} - \underline{s}_k \|^2}{2\sigma^2} \tag{45}$$

If the squared length of a vector $\underline{x}$ is written as $\| \underline{x} \|^2$, then

$$\| \underline{x} \|^2 = \sum_{i=1}^{I} x_i^2 \tag{46}$$

and similarly, $\| \underline{z} - \underline{s}_k \|^2$ represents the distance squared between $\underline{z}$ and $\underline{s}_k$. Substitution of Eq (45) into Eq (42) yields the following

$$P\{m_j\} \left[(2\pi\sigma^2)^{-I/2} \exp - \frac{\| \underline{z} - \underline{s}_j \|^2}{2\sigma^2} \right]$$

$$= \max_k P\{m_k\} \left[(2\pi\sigma^2)^{-I/2} \exp - \frac{\| \underline{z} - \underline{s}_k \|^2}{2\sigma^2} \right] \tag{47}$$

Initially, we consider the equally likely case, hence $P\{m_j\} = P\{m_k\}$. Cancelling common terms reduces Eq (47) to

$$\exp \frac{-\| \underline{z} - \underline{s}_j \|^2}{2\sigma^2} = \max_k \exp \frac{-\| \underline{z} - \underline{s}_k \|^2}{2\sigma^2} \tag{48}$$

If we take the logs and cancel common terms, the result is

$$-\| \underline{z} - \underline{s}_j \|^2 = \max_k -\| \underline{z} - \underline{s}_k \|^2 \tag{49}$$

Finally, multiplying both sides by minus 1 causes the maximum to become a minimum and Eq (49) is now

$$\| \underline{z} - \underline{s}_j \|^2 = \min_k \| \underline{z} - \underline{s}_k \|^2 \tag{50}$$

Picturing the messages as points in an I-dimensional space, the region Z_j is seen to be the set of all points that are closer to s_j than to any other signal point.

Consider a two-dimensional signal space with two equally likely signal points. From the above, we can visualize this space being divided in half, i.e., all the points closer to the first point in one half plane, and the remaining points which are closer to the second point in the other half plane. The dividing line or boundary is seen to be the perpendicular bisector of the line connecting the two signal points. In fact, for any two-dimensional signal space, all the decision regions will be bounded by such segments. It is interesting to note that the boundary will only be a bisector when the signals are equally likely. When the signals are not equally likely, the boundary will still be perpendicular, but will be shifted toward the signal with lesser probability. Another point to note is that the location of the boundary (hence, the decision rule) has been completely specified without using the variance of the noise. This is so only because the signals are equiprobable. Although finding the decision regions for equiprobable signals is straightforward, determining the probability of error can be more complex.

Probability of Error for Multiple Decisions

The probability of error given message m_k, associated with a given decision, can be expressed as 1 minus the probability of a correct decision. That is:

$$P\{e|m_k\} = 1 - P\{\text{Correct Decision}\}$$

$$= 1 - p(d_k|m_k)\ P\{m_k\} \qquad (51)$$

Substitution of Eq (9) into Eq (51) gives the probability of error given m_k as,

$$P\{e|m_k\} = 1 - \int_{Z_k} p(\underline{z}|m_k)\ d\underline{z} \cdot P\{m_k\} \qquad (52)$$

The simplicity of this equation is deceiving because in general it requires the computation of an I-fold integral. It would be foolhardy to attempt to calculate this with pencil and paper for all but the most trivial of problems. However, as will be seen, the developed program uses this equation to provide an approximation of the conditional probability of error. When the message is m_k, an error will occur if z is not in Z_k. Thus

$$P\{e|m_k\} = P\{\underline{z} \notin Z_k|m_k\} \qquad (53)$$

It is the direct calculation of Eq (53) which provides the aforementioned complexity. In general, the error event $(P\{\underline{z} \notin Z_k|m_k\})$ becomes a function of inseparable joint densities which cannot be simplified or easily computed.

Hence, it is not always possible to obtain meaningful exact expressions for error probabilities.

Concept of the Union Bound

A more useful approach is to find an upper bound of the probability of error that is simple and that is also a good approximation. We desire an upper bound rather than simply a good approximation because systems are generally designed to meet some *minimum performance standard*. The designer must be certain that a probability of error, for example, is no larger than some given number. If it can be guaranteed that an upper bound on the error probability is lower than this number, then the designer knows the requirement has been met. Using an approximation, the designer could never be sure.

Consider the probability of the union of two events A and B as given by

$$P(A \cup B) = P(A) + P(B) - P(A \cap B) \tag{54}$$

Assume $P(A)$ and $P(B)$ are known, but not $P(A \cap B)$. Since $P(A \cap B)$ must always be greater than zero, if it is deleted from Eq (54), the approximated value of $P(A \cup B)$ will always be greater than the exact value of $P(A \cup B)$. This, then, is the concept of the upper bound. The amount the approximation of $P(A \cup B)$ exceeds the exact value is, of course, the value of $P(A \cap B)$.

This approach can be generalized quite easily. It is called the "Union Bound" and is stated in the following theorem (Ref 3:115):

If the event A is the union of K events

$$A = \bigcup_{k=1}^{K} E_k$$

then the probability of A is bounded by

$$P(A) \le \sum_{k=1}^{K} P(E_k) \tag{55}$$

Using Eq (50), we can write the decision region Z_k as

$$Z_k = \{\underline{z} : \| \underline{z} - \underline{s}_k \|^2 \le \| \underline{z} - \underline{s}_\ell \|^2\} \tag{56}$$

for all $\ell \ne k$, which can be written

$$Z_k = \bigcap_{\substack{\ell=1 \\ \ell \ne k}}^{K} \{z : \| \underline{z} - \underline{s}_k \|^2 \le \| \underline{z} - \underline{s}_\ell \|^2\} \tag{57}$$

Additionally, its compliment, Z_k^c, can be expressed

$$Z_k^c = \bigcup_{\substack{\ell=1 \\ \ell \ne k}}^{K} \{z : \| \underline{z} - \underline{s}_k \|^2 \ge \| \underline{z} - \underline{s}_\ell \|^2\} \tag{58}$$

Using this, the conditional probability of error is

$$P\{e|m_k\} = \Pr\{z \in Z_k^c|m_k\} \tag{59}$$

By substitution of Eq (58) into Eq (59), we get Eq (60):

$$P\{e|m_k\} = \Pr\left[z \in \bigcup_{\substack{\ell=1 \\ \ell \neq k}}^{K} \{\underline{z} : \|\underline{z} - \underline{s}_k\|^2 \geq \|\underline{z} - \underline{s}_\ell\|^2\} \;|m_k \right] \tag{60}$$

Applying the concept of the Union Bound, this becomes

$$P\{e|m_k\} \leq \sum_{\substack{\ell=1 \\ \ell \neq k}}^{K} \Pr\{\underline{z} : \|\underline{z} - \underline{s}_k\|^2 > \|\underline{z} - \underline{s}_\ell\|^2|m_k\} \tag{61}$$

Each term in the summation can be simplified further. When the message is m_k, $\underline{z} = \underline{s}_k + \underline{n}$. Thus, the expression for each ℓ becomes

$$\Pr\{\|\underline{z} - \underline{s}_k\|^2 > \|\underline{z} - \underline{s}_\ell\|^2|m_k\}$$

$$= \Pr\{\|\underline{n} + \underline{s}_k - \underline{s}_k\|^2 > \|\underline{n} + \underline{s}_k - \underline{s}_\ell\|^2\} \tag{62}$$

which can be reduced to yield

$$\Pr\{\|\underline{z} - \underline{s}_k\|^2 > \|\underline{z} - \underline{s}_\ell\|^2|m_k\}$$

$$= \Pr\{2\underline{n}^T(\underline{s}_\ell - \underline{s}_k) > \|\underline{s}_\ell - \underline{s}_k\|^2\} \tag{63}$$

As shown in Appendix A, the term $2\underline{n}^T(\underline{s}_\ell - \underline{s}_k)$, is a zero mean gaussian random variable with variance $4\sigma^2\|\underline{s}_\ell - \underline{s}_k\|^2$. Using this fact provides the following simplified expression

for the probability. That is,

$$\Pr\{\| \underline{z} - \underline{s}_k \|^2 > \| \underline{z} - \underline{s}_\ell \|^2 | m_k\} = Q\left[\frac{d_{\ell k}}{2\sigma}\right] \tag{64}$$

where $d_{\ell k}$ is the distance between any two message vectors $\underline{s}_\ell$ and $\underline{s}_k$,

$$d_{\ell k} = \| \underline{s}_\ell - \underline{s}_k \| \tag{65}$$

Hence, the conditional probability of error can be bounded by

$$P\{e|m_k\} \le \sum_{\substack{\ell=1 \\ \ell \ne k}}^{K} Q\left[\frac{d_{\ell k}}{2\sigma}\right] \tag{66}$$

To determine the overall average probability of error, Eq (66) is averaged over all K so that the Union Bound for the system becomes

$$P_e \le \sum_{k=1}^{K} \frac{1}{K} \sum_{\substack{\ell=1 \\ \ell \ne k}}^{K} Q\left[\frac{d_{\ell k}}{2\sigma}\right] \tag{67}$$

From Eqs (66) and (67), we see that the Union Bound, an approximation to the probability of error, requires only that the distances, $d_{\ell k}$, between the signals, and σ, the system noise energy be known. The accuracy of this approximation may be subject to question, but as a design tool, it is certainly much easier to compute and implement than Eqs (52) or (53).

However, we have only considered the equally likely case. The extension to the general case is not difficult and the development quite similar. The general decision rule given by Eq (42) says to select the message for which $P\{m_j\}\ p(\underline{z}|m_j)$ is a maximum. If we do not assume that the [illegible] are the same, they will not cancel and will appear in the final definition of the decision region Z_j.

If $P\{m_j\}\ p(\underline{z}|m_j)$ is to be the maximum of a set, clearly it must be larger than every other element in the set. Therefore, the set of $\underline{z}$ for which $P\{m_j\}\ p(\underline{z}|m_j)$ is a maximum can be written as the joint intersection of those regions of Z for which it is larger than $P\{m_\ell\}\ p(\underline{z}|m_\ell)$ for each $\ell \neq j$. In other words,

$$Z_j = \bigcap_{\substack{\ell=1 \\ \ell \neq k}}^{K} \{\underline{z} : P\{m_j\}\ p(\underline{z}|m_j) > P\{m_\ell\}\ p(\underline{z}|m_\ell)\} \tag{68}$$

If we use $Z_{j\ell}$ to denote the ℓ^{th} term on the right of Eq (68), we find that for the gaussian case

$$Z_{j\ell} = \left\{ \underline{z} : P\{m_j\} \exp \frac{-\| \underline{z} - \underline{s}_j \|^2}{2\sigma^2} > P\{m_\ell\} \exp \frac{-\| \underline{z} - \underline{s}_\ell \|^2}{2\sigma^2} \right\} \tag{69}$$

Taking logs and cancelling common terms simplifies the expression and it becomes

$$Z_{j\ell} = \left\{ \underline{z} : \| \underline{z} - \underline{s}_j \|^2 < \| \underline{z} - \underline{s}_\ell \|^2 + 2\sigma^2 \ln \frac{P\{m_j\}}{P\{m_\ell\}} \right\} \quad (70)$$

If $P\{m_\ell\}$ and $P\{m_j\}$ are equal, the $Z_{j\ell}$ is just the set of points that are closer to $\underline{s}_j$ than to $\underline{s}_\ell$. If $P\{m_\ell\}$ is smaller than $P\{m_j\}$, some of the points that are closer to $\underline{s}_\ell$ will be included in $Z_{j\ell}$.

We know that for equal probabilities the boundary of $Z_{j\ell}$ is the perpendicular bisector of the line that joins $\underline{s}_\ell$ and $\underline{s}_j$. As mentioned previously, even if the probabilities are not equal, the boundary is still perpendicular to $(\underline{s}_\ell - \underline{s}_j)$. To show this, Eq (70) is rewritten by writing out the squared vector lengths and collecting similar terms:

$$Z_{j\ell} = \left\{ \underline{z} : 2\underline{z}^T\underline{s}_\ell - 2\underline{z}^T\underline{s}_j < \| \underline{s}_\ell \|^2 - \| \underline{s}_j \|^2 + 2\sigma^2 \ln \frac{P\{m_j\}}{P\{m_\ell\}} \right\} \quad (71)$$

Hence, the boundary of the region is the set of $\underline{z}$ for which

$$\underline{z}^T(\underline{s}_\ell - \underline{s}_j) = \frac{1}{2}\| \underline{s}_\ell \|^2 - \frac{1}{2}\| \underline{s}_j \|^2 + \sigma^2 \ln \frac{P\{m_j\}}{P\{m_\ell\}} \quad (72)$$

The set of $\underline{z}$ for which $\underline{z}^T(\underline{s}_\ell - \underline{s}_j)$ is a constant is a line that is perpendicular to $(\underline{s}_\ell - \underline{s}_j)$.

As for the equiprobable case, the probability of error typically is not easy to calculate. However, the union bound can be used on $Z_{j\ell}^c$, and the error probability is bounded by:

$$P_e \leq \sum_{j=1}^{K} \frac{1}{K} \sum_{\substack{\ell=1 \\ \ell \neq j}}^{K} \Pr\left\{ \|\underline{n}\|^2 > \|\underline{n} + \underline{s}_j - \underline{s}_\ell\|^2 + 2\sigma^2 \ln \frac{P\{m_j\}}{P\{m_\ell\}} \right\} \tag{73}$$

Using the same logic as in the equiprobable case, this becomes

$$P_e \leq \sum_{j=1}^{K} P(m_j) \sum_{\substack{\ell=1 \\ \ell \neq j}}^{K} Q\left[\frac{d_{\ell j}}{2\sigma} - \frac{\sigma}{d_{\ell j}} \ln\left(\frac{P\{m_j\}}{P\{m_\ell\}} \right) \right] \tag{74}$$

For a particular message m_k, the conditional probability of error can thus be expressed as

$$P\{e|m_k\} < \sum_{\substack{\ell=1 \\ \ell \neq k}}^{K} Q\left[\frac{d_{\ell k}}{2\sigma} - \frac{\sigma}{d_{\ell k}} \ln\left(\frac{P\{m_\ell\}}{P\{m_k\}} \right) \right] \tag{75}$$

As noted previously, it is not possible to fully specify the decision regions without knowing σ^2. If σ^2 is large, the decision will be highly biased in favor of the more probable messages. This is reasonable since for large

σ^2 the received signal is not very reliable. On the other hand, if σ^2 is small, the a priori probabilities do not affect the decision regions very much. Equation (75) is the relationship used to compute the Union Bound on the probability of error in the developed program.

III. Concept and Design of Software

Introduction

As a tutorial tool, the main objective of Program SIGDET is to acquaint the user with certain aspects of communication signal set design and with how the selection of the signal set affects the operation of the communication system. By relieving the user of hours of tedious numerical calculations, SIGDET allows him to alter signal location (energy), proximity, probability, and system noise to quickly gain an appreciation of the interaction and relationships of each of these factors.

The program consists of 23 subroutines in all; ten are designed to allow the user to input and manipulate data, nine are strictly computational, three are dedicated to outputting or displaying data, and one subroutine, though primarily computational, also provides output.

In general, once the signal set has been specified, practically all required computations are carried out using default values for all other parameters. This is done prior to the option prompt being provided to the user. Hence, all options are available at all times. Should the signal set be changed (translated, having a point added or deleted, etc.) or should a previously defaulted variable parameter be user specified (system noise energy), all required computations are again performed before the option prompt is provided.

After an explanation of the main program operation, each subroutine will be discussed and, where necessary, the algorithms used will be explained. The data input routines will be presented first, followed by the computational routines, and finally the routines which display the answers. [illegible] the matrix which stores the coordinates of the bisector intersections is helpful for following the construction and execution of the program. An explanation of this matrix is therefore included at the end of the chapter.

Main Program Operation

With the signal set entered and the values of system noise, scaling factor, and signal probabilities assigned by default, the computation below occurs:

1. Using only the number and coordinates of the signal points, the energy in each signal is determined.

2. Using the data above and the system noise energy (default value gives a noise variance or PSD = 1.0), the signal-to-noise ratio of each signal is determined.

3. Using the signal coordinates and the signal probabilities (default values are equally likely), the system center of gravity is computed.

4. Now the computational heart of SIGDET is performed. The minimum and maximum X-Y dimensions of the signal set are determined. If the X-Y axis is not within these limits, adjustments in the required directions are made to

accommodate both axes. Then the equation of an imaginary circle which circumscribes the entire signal set is determined using these dimensions. Next, the signals are considered two at a time, the distance between each set of two points is computed and the midpoint of the line drawn between them determined. Using this midpoint and the point-slope form of the equation of a straight line, an equation for the perpendicular bisector between each two-point set is determined. Simultaneous solution of this equation and the equation of the circle yields the two points of intersection of each perpendicular bisector and the circle, that is, the endpoints of each bisector. These coordinate pairs are stored in a large matrix and become the basis for all future computation. Using the two-point form of the equation of a straight line, the points of intersection of all the perpendicular bisectors are determined. Another subroutine determines which of these points of intersection are the endpoints of the line segments which make up the decision boundary regions.

5. Finally, using the computed signal separation, the system noise energy, and the probabilities of each signal, an upper bound on the probability of error for each signal is determined and the program is ready to prompt for user options.

Data Input Subroutines

The table below lists each data input subroutine and its particular function.

TABLE I

DATA INPUT SUBROUTINES

Subroutine Name	Function
CURSOR	Input signal points using cursor
DIRECT	Input signal points by specifying coordinate pairs
TRANS	Input amount of movement in X-Y directions signal set is to be moved
ROTATE	Input angle of rotation
ADDGRA	Add signal point using cursor
ADDDIR	Add signal point by specifying coordinates of the point
DELETE	Remove unwanted signal point
SNOISE	Specify system noise energy
SGPROB	Specify probability of each signal
SCALER	Input scaling factor to expand field of display

Subroutine CURSOR. The function of this subroutine is to allow the user to input the signal set coordinates by utilization of a graphical cursor. A square region is drawn on the screen, the X-Y axis displayed, and the cursor (represented by crosshairs) provided which is to be manipulated by user. In order for the graphics package to be able to assign x and y values to any location of the cursor, some frame of

reference or scale must be established so that the cursor "knows where it is." For this reason, the user is required to provide the X and Y dimensions of the signal set he wishes to study. These dimensions thus describe the "window" inside which the entire signal set will reside. The user positions the intersection of the crosshairs, called the crosspoint, at the location he desires to place a signal point. By entering the letter "P" via the terminal keyboard, the user identifies the cursor location as a signal point and the coordinates of this location are computed and stored in matrix "PTS." An asterisk is displayed at the location of the point and the number of the signal is displayed to the right of the asterisk. The cursor is then free to be moved to the next signal location. A count is maintained of the number of signals so entered and when the entire signal set has been input, the cursor is moved to a position just below the window and a listing of the signal set coordinates is displayed. Since there is limited space beneath the window, only the first four points are listed and the user is given the choice of having the screen cleared and the remaining points listed, or simply continuing with the program. In either case, the prompt causes a pause in program execution to allow the user to copy screen display if desired prior to continuing. After the decision has been made and executed, control of the program returns to mainline.

<u>Subroutine DIRECT</u>. The function of this subroutine is to allow the user to input the signal set points simply by specifying the actual coordinate pair of each signal point. This enables accurate placement of signals, particularly for symmetric signal sets and the use of a terminal [illegible] ical [illegible] lity [illegible] the routine provides a prompt for the user to input the coordinates of each signal. The standard FORTRAN READ routine is used to receive input from the user, i.e., the user needs only to enter the coordinates in normal decimal notation. The values are separated by a comma and a carriage return signals completion of a line of input. After all coordinate pairs have been entered, the program execution begins.

<u>Subroutine TRANS</u>. This subroutine provides the capability of translating the signal set. The user is given a prompt requesting the amount of movement in the X and Y directions. In order to translate the signal set, the amount of movement in the X direction, DELTAX, is added to the x-coordinate of each signal point. Similarly, DELTAY is added to the y-coordinate of each signal point. In this way the original frame of reference, that is, the original X-Y axis remains fixed and each signal (hence the entire signal set), is translated in relation to it. After all the signal point coordinates have been appropriately adjusted, control returns to mainline.

Subroutine ROTATE. Subroutine ROTATE allows the signal set to be rotated about the X-Y axis. The routine begins by asking the user to provide the angle of rotation, positive or negative between 0 and 360 degrees. This angle in degrees is immediately converted to radians and a check [illegible] as [illegible] not greater [illegible] radians. If so, an error message is displayed and a prompt for the angle is again provided. In order to perform the rotation, the values of all signal point coordinate pairs are changed to represent their *new* value in the rotated system. The fundamental formulae for rotation of an X-Y axis through an angle θ are given by:

$$x = x'\cos\theta - y'\sin\theta$$

$$y = x'\sin\theta + y'\cos\theta \qquad (76)$$

where x' and y' are the coordinates of point (x,y) in the *rotated* X-Y axis. Rewriting these equations in terms of x' and y' yields:

$$x' = y\cos\theta - x\sin\theta$$

$$y' = x\cos\theta + y\sin\theta \qquad (77)$$

These then are the equations used to determine the new values of the signal point coordinates in relation to an X-Y *axis which has been rotated* an amount equal to theta. However, in order to represent the values when the *signal set is rotated* instead, the sign of the user inputted angle is

changed prior to computation. Furthermore, the two equations interact, i.e., once a new value of the signal's x-coordinate is computed, that value cannot be used to compute the new y-coordinate. The old x-coordinate value must still be available. Hence, prior to any computation, the [illegible] matrix ITP, [illegible] named "SAVPTS" so that the original signal set is continuously available when computing the new signal set values. When all points have been thus rotated, control returns to mainline.

<u>Subroutine ADDGRA</u>. Should the user desire to add a signal, he has two methods available. This particular routine allows the additional signal to be added graphically. To accomplish this, the user is first provided a display of the current signal set. The display is presented via a call to subroutine PLOT (see Section III, Output Subroutines). Subroutine PLOT displays the window, the X-Y axis, the current signal set, and a prompt providing the user the choice of having the screen cleared and the coordinates of the signals listed or continuing with program execution leaving screen display intact. The second choice is mandatory, whereupon the cursor crosshairs will be displayed. The user positions the crosspoint as desired and enters a "P" via the terminal keyboard. As before, an asterisk and corresponding signal number are displayed. Finally, the cursor is moved to the lower portion of the screen and the coordinates of the added

signal are listed on the screen. Since the user may wish to copy this new signal set as displayed, program execution is halted and a prompt for user to enter any digit to continue is provided. After a digit is entered, screen is cleared and execution returns to mainline.

Subroutine [illegible]. In conjunction with subroutine ADDGRA, this subroutine allows the user to add additional signal points by directly specifying the coordinates of the new signal. It is very similar to subroutine DIRECT; however, the prompts to the user have been altered. Here the user is initially prompted for the x-coordinate of the new signal and then prompted for the y-coordinate. The input format is the same as in DIRECT and after both coordinates are entered, program control returns to mainline.

Subroutine DELETE. This subroutine, as may be expected, allows the user to remove an unwanted signal. Since each signal is numbered, the user simply provides the number of the unwanted signal. This number, given the variable name GONE, is used as an index variable in a do loop that erases the coordinates of the deleted signal and successively "moves up" each remaining signal in the signal point matrix PTS. Hence the user should be aware that all points with a number greater than the one deleted will be renumbered.

Subroutine SNOISE. This subroutine has a two-fold function. It informs the user what the system noise energy

would have to be in order for the largest signal-to-noise ratio to be zero decibels. This provides the user a starting point for selecting variable noise energy values. Secondly, this routine allows the user to override the default noise energy of 2.0 (which results in a noise variance or power [illegible]

By utilization of the intrinsic function MAX, the largest individual signal energy is determined. Then from the definition for signal-to-noise ratio (SNR) (Ref 10:250),

$$\left.\frac{E_s}{N_o}\right|_{db} \triangleq 10 \log_{10} \frac{E_s}{N_o} \tag{78}$$

it is evident that when $N_o = E_s$, the signal-to-noise ratio will be zero db. Hence the user is informed that the noise energy level must be equal to this maximum signal energy for the condition above to exist. After the user supplies the desired noise energy, a check is made to assure that the entered value is greater than zero. If not, an error message is displayed and prompt provided again. The routine informs the user of the resulting value of the noise variance or power spectral density prior to returning to mainline.

<u>Subroutine SGPROB</u>. The purpose of this routine is to allow the user to specify the individual signal probabilities. The program automatically computes all signal probabilities for the equally likely case. Upon selecting this option, the user is provided with a prompt and enters each

signal probability one at a time. Standard format is used, and as each probability is read it is checked to certify that it is greater than zero. If not, an error message is provided and the prompt again given. After all probabilities have been entered, a check is made of their sum. Should the [illegible] than 0.95 [illegible] 1.05, an error message is displayed and all probabilities must be reentered. After a more accurate set of signal probabilities has been entered, the program returns to mainline.

<u>Subroutine SCALER</u>. Subroutine SCALER enables the user to alter the "size" of the "window" encompassing the signal set. The signals, of course, maintain their relative positions, but the scale or the boundary which surrounds them can be made to expand, thus in effect compressing the display of the signal set. This is useful in that it allows all boundary region intersections to be displayed; in fact, it is used by subroutine EXACT to do just that. The window can also be reduced, but the author cannot envision a situation in which a window smaller than the original would be needed. Subroutine WINDOW is structured such that it determines the minimum and maximum X and Y dimensions so that the entire signal set is enclosed, the X-Y axis is included within the window, and the window is a square. The window is made a square to ensure that the scales along the X and Y axis are identical (see subroutine WINDOW). Once the values of XMIN, XMAX, YMIN, and YMAX have been determined, the

center of the square is computed and then these four values are recomputed with the scaling factor as a multiplier. The default value is one; a value greater than one expands the window, whereas a value less than one would shrink the window. In conclusion, the actual physical dimension of the [illegible] along the axes have been altered.

The Computational Subroutines

Once the required data has been provided, the program is ready to perform the basic calculations. Table II lists each computational subroutine used along with a description of its function.

Subroutine SGENGY. The function of this routine is to compute the individual energy of each signal and then store the results in array "ENGY". Given the x and y-coordinates of a signal point on a two dimensional plane, it computes the energy as the "length" or norm of the vector from the origin to the signal point. Hence the energy is the sum of the square of the X and Y distances. Mathematically this becomes:

$$\text{Energy} = (\text{x-coordinate})^2 + (\text{y-coordinate})^2 \qquad (79)$$

After each signal energy is computed and stored, control returns to the main program.

Subroutine SNRCOM. Subroutine SNRCOM uses the system noise energy level and the computed individual signal

TABLE II

COMPUTATIONAL SUBROUTINES

Subroutine Name	Function
SGENGY	Compute energy of individual signals
[illegible]	[illegible]-to-noise ratio of each signal
GRACEN	Compute center of gravity of signal set
WINDOW	Determines minimum and maximum X and Y dimensions of signal set and equation of circumscribing circle
BISECT	Determines coordinates of endpoints of all perpendicular bisectors
POINTS	Determines coordinates of all points of intersection of all perpendicular bisectors
REGION	Determines which points above are the endpoints of the decision region line segments
PERROR	Uses the concept of the Union Bound to compute an upper limit to the probability of error for each signal
EXACT	Computes the probability of error for each signal by integrating the density function providing a tighter bound
CMPUTE	Forms the basis for computation and display of the decision regions. Calls WINDOW, BISECT, POINTS, and REGION

energies to determine the signal-to-noise ratio (SNR) for each signal. The computation is based on the definition for the signal-to-noise ratio used in subroutine SNOISE:

$$\left.\frac{E_s}{N_o}\right|_{db} \triangleq 10 \log_{10} \frac{E_s}{N_o} \tag{78}$$

Since the $\log_{10}$ of zero is undefined, when a signal is at the origin or has zero energy, its SNR should also be undefined. In this routine, however, if a signal has zero energy, its SNR is automatically set equal to -999.9999 to indicate this situation. Clearly, if a signal has no energy, there is in fact no signal and the receiver sees only noise. After all the SNR's have been computed and stored in array "SNR" program control returns to mainline.

Subroutine GRACEN. This subroutine computes the physical center of gravity of the signal set based on the location of the signals and their probabilities. When the signal probabilities are equally likely, the x-coordinate of the resulting center of gravity is easily seen to be just the average of the sum of the x-coordinates of all the system's signals. The same is true for the y-coordinate. This is not the case when the signal probabilities are no longer identical. Borrowing a little theory from the physics of masses, an expression for the moment of inertia or mean energy around the origin of a system of N point masses is given by (Ref 10: 247):

$$\overline{E_m} \triangleq \sum_{i=1}^{N} P\{m_i\}E_i = \sum_{i=1}^{N} P\{m_i\}||\underline{s}_i||^2 \tag{80}$$

where the mass of the i^{th} point is $P\{m_i\}$ and its position is $\underline{s}_i$. For a given set of point masses (signal probabilities), the mean energy can be minimized, without affecting the probability of error, by subtracting from each signal $\underline{s}_i$, a constant $\underline{a}$ such that:

$$\sum_{i=1}^{N} P\{m_i\}||\underline{s}_i - \underline{a}||^2 \tag{81}$$

is a minimum. Since the moment of inertia (mean energy) is a minimum when taken around the centroid (center of gravity), it follows that $\underline{a}$ should be chosen so that the resulting centroid coincides with the origin. Thus, $\underline{a}$ or the center of gravity can be found from the relation:

$$\underline{a} = \sum_{i=1}^{N} P\{m_i\}\underline{s}_i \tag{82}$$

Hence, in order to determine the x-coordinate of the system center of gravity, this routine sums the product of each signal's probability and its x-coordinate. Similarly, the y-coordinate is found. After all the signals have been considered, the center of gravity has been determined and control of the program returns to mainline.

<u>Subroutine WINDOW</u>. The purpose of subroutine WINDOW is to determine the X and Y dimensions of the graphical

window such that the entire signal set is enclosed and the X-Y axis is within the confines of its boundaries. Additionally, this routine determines the center of the window and from this computes the radius of a circle circumscribing the square window. In order to determine the window dimensions, the minimum and maximum x and y coordinate values of the entire signal set are found. The range of each dimension is computed. Then a check is made to see if the Y-axis is within the X-dimension range. Basically, if the maximum x-coordinate value is greater than zero and if the minimum x-coordinate value is less than zero, then the Y-axis is within the X-dimension range. If this is not the case, then an adjustment to the appropriate X-dimension is made. For instance, if all the signal points have x-coordinate values greater than zero, then the minimum X-dimension will be adjusted so that it is negative and the X-range will thus include the Y-axis. The amount that the X-dimension is made negative, i.e., the value of the adjusted XMIN, is one-tenth of the initial value of X-range. A similar check and adjustment is made for the y-coordinate values. The signals themselves always remain unchanged, however. Next, the window is squared so that the scales along each axis will be identical. In order to accomplish this, the difference in length of the two ranges is determined. The minimum value of the shorter range has one-half this difference subtracted from it, whereas the maximum value of the shorter range has

one-half the difference added to it. Hence the shorter range is lengthened on each end so that it now equals the longer range. With the window geometrically as well as physically square, the new range of each dimension is determined and the center of the window reverified. At this [illegible] an[illegible] ion is adjus[illegible] according to the scaling factor. With the window still square, the adjusted X and Y ranges are recomputed. Finally the radius of the circumscribing circle is determined as the length from the center of the window to one of the corners. To preclude roundoff error, a fudge factor of one-half X-range is added to the radial length. Thus, the window scale can be adjusted by the user, but it always remains a square and its physical size as displayed on the screen will always be the same.

Subroutine BISECT. This subroutine determines the coordinates of the endpoints of the perpendicular bisectors between all combinations of any two signal points. As mentioned in the matrix "IN" discussion, the perpendicular boundary line between any two signal points will be a bisector only when the probabilities of the two points are equal; otherwise, the boundary will shift toward the signal with the smaller probability. Although this subroutine is designed to work with any combination of legal probabilities which sum to unity, the term bisector will be used regardless of the actual location of the perpendicular boundary.

This subroutine is called by subroutine CMPUTE, hence it is called once for every signal point. The points are indexed successively and BISECT takes the point selected by subroutine CMPUTE (hereafter to be called the lead point) and considers it with every other signal point. First the distance between the two signal points under consideration is determined and stored in matrix "DISTA". Then the midpoint of the line connecting the two signal points is found. Using the point-slope form of the equation of a straight line, the coordinates of the midpoint, and the slope of the connecting line, the equation of each bisector is known. A check is made to see if the bisector is horizontal, for if it is, then the endpoints are readily available, i.e., the x-coordinates will be just the minimum and maximum X-dimensions of the window and the y-coordinates will be equal to the y-coordinate value of the midpoint. Similarly, a check is made to determine if the bisector is vertical with correspondingly easily obtained endpoints. Otherwise, the endpoints must be found from the solution to a set of simultaneous equations, the equation of the bisector and that of the circumscribing circle. The details of these computations will be presented below. The coordinates of the points of intersection thus become the coordinates of the endpoints of each bisector and are stored in matrix IN. These coordinates become the basis for all remaining computation concerned with the decision region boundaries.

Discussion of Bisector Location. As mentioned above, the location of a perpendicular bisector is at the midpoint of a line connecting two signal points only when the two signals are equally likely. If this is not the case, the boundary is shifted by an amount "DELTA", found from the following equation (Ref 10:251):

$$\Delta_{ij} = \frac{N_o}{2d} \ln \frac{P\{m_i\}}{P\{m_j\}} \tag{83}$$

where N_o = system noise energy

d = distance between the two signals

$P\{m_k\}$ = probability of signal k

and where the sign of DELTA is determined by the $\ln \frac{P\{m_i\}}{P\{m_j\}}$. That is, when $P\{m_i\} > P\{m_j\}$, the midpoint will be shifted toward m_j. This subroutine uses trigonometric relationships to determine the adjusted coordinates of each "midpoint" using the value of DELTA computed for each pair of signals considered.

For example: Given, Point 1 (1,1) with $P\{m_1\} = .8$

Point 2 (3,2) with $P\{m_2\} = .2$

σ, defined as the angle of inclination of the connecting line (not to be confused with the symbol for noise energy)

Figure 2 shows the bisector for the equally likely case and Figure 3 shows how the bisector is shifted due to the change in the signal probabilities.

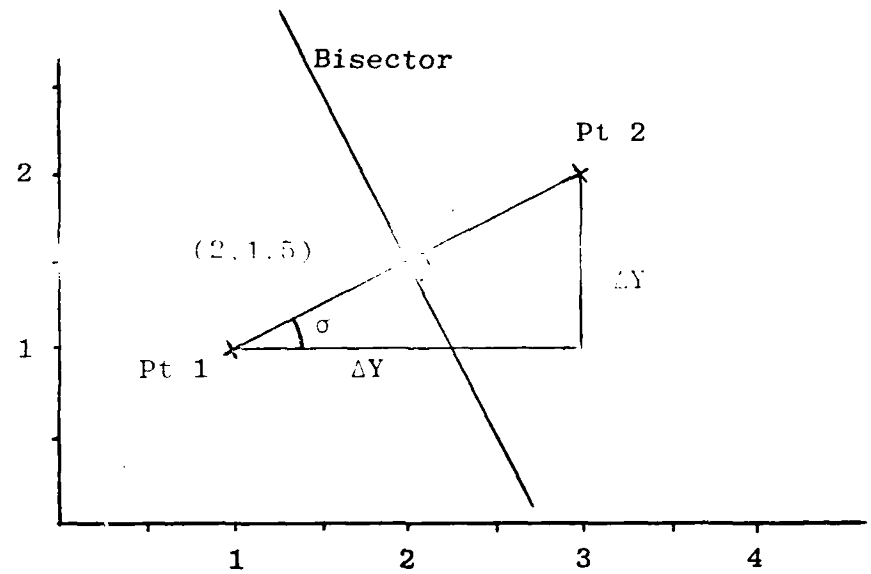


Fig 2. Bisector for Equiprobable Signals

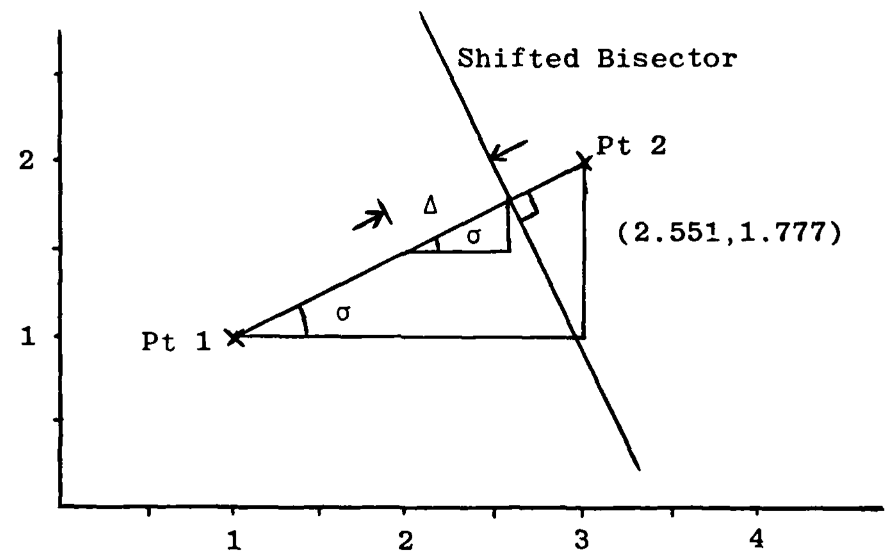


Fig 3. Bisector for Signals Not Equiprobable

The new midpoint location is computed as follows:
First the value of SIGMA is found and then the value of DELTA is determined.

$$\sigma = \arctan \frac{\Delta y}{\Delta x} = \arctan \frac{1}{2} = 26.56°$$

$$\Delta = \frac{2}{2 \cdot \sqrt{5}} \cdot \ln \frac{P(m_1)}{P(m_2)} = \frac{1}{\sqrt{5}} \ln 4 = 0.620$$

(Assuming noise energy, N_o, is taken as 2.0)
From the figures and basic trigonometry, the new coordinates become:

$$xnew = xold + \Delta\cos\sigma = 2. + 0.554 = 2.554$$

$$ynew = yold + \Delta\sin\sigma = 1.5 + 0.277 = 1.777$$

In such a fashion, the subroutine shifts the "midpoint" depending on the system noise and the signal probabilities.

Unfortunately, determining the value of SIGMA is not as straightforward as it appears on the surface. The FORTRAN intrinsic function ATAN returns a value between $-\frac{\pi}{2}$ and $\frac{\pi}{2}$ for the arctangent. Hence, if in the figures above, σ were obtuse, this function would not provide the positive angle greater than $\frac{\pi}{2}$, but its negative supplement. Therefore, after σ is determined, two situations may exist:
(1) its value is returned as positive, i.e., between zero and $\frac{\pi}{2}$, or (2) as negative, i.e., less than zero but greater than $-\frac{\pi}{2}$. BISECT checks for this and then must determine in

which order the coordinates of the signal points happened to be used to compute the value of σ. The two cases below demonstrate this.

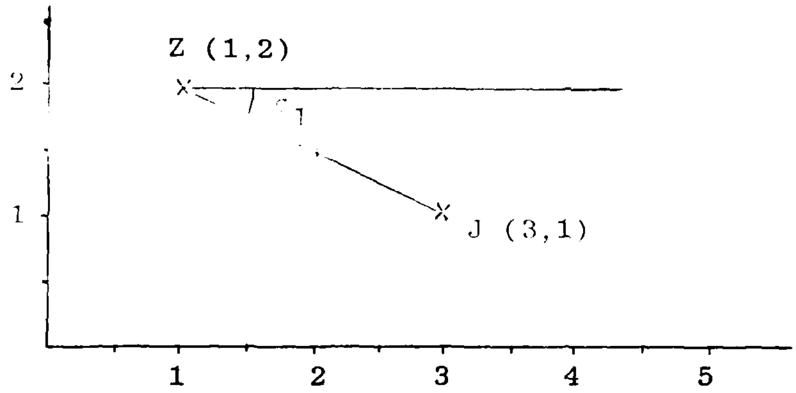


Fig 4. Computation of Angle of Inclination, Case 1

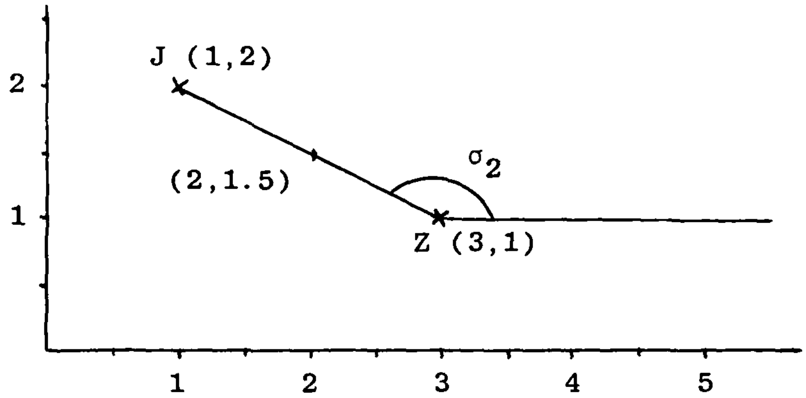


Fig 5. Computation of Angle of Inclination, Case 2

Assume the lead point is point Z, and the next point is J. We would then compute SIGMA from the following equation:

$$\sigma = \arctan \frac{y_j - y_z}{x_j - x_z} \qquad (84)$$

Hence, in Figure 4 we have:

$$\sigma_1 = \arctan \frac{1-2}{3-1} = \arctan(-\tfrac{1}{2})$$

$$\sigma_1 = -26.5°$$

which is the desired angle. In Figure 5, we compute:

$$\sigma_2 = \arctan \frac{2-1}{1-3} = \arctan(-\tfrac{1}{2})$$

$$\sigma_2 = -26.5°$$

which is not the desired angle, but its supplement! BISECT gets around this by checking whether the y-coordinate of the lead point (Z) is less than that of the second point (J). There are then four conditions which may exist:

1. SIGMA is greater than or equal to zero and
   a. Z's y-coordinate is greater than J's or
   b. Z's y-coordinate is less than J's

or 2. SIGMA is less than zero and
   a. Z's y-coordinate is greater than J's or
   b. Z's y-coordinate is less than J's

BISECT checks to determine which situation exists and only then computes the adjusted location of the "midpoint". Four different sets of two equations are used to correctly adjust the x and y-coordinates. In the example above, let us assume that Figure 5 indicates the actual locations of the lead point and the next point under consideration. Then, since the returned value of SIGMA is less than zero and Z's

y-coordinate is less than J's y-coordinate, the equations would become:

$$xnew = xold - \Delta\cos\sigma$$

$$ynew = yold - \Delta\sin\sigma$$

If we further assume $P\{m_z\} = 0.8$ and $P\{m_j\} = 0.2$, then DELTA becomes:

$$\Delta = \frac{N_o}{2d} \ln \frac{P\{m_z\}}{P\{m_j\}} = \frac{2}{2\cdot\sqrt{5}} \ln 4 = 0.6199$$

and hence:

$$xnew = xold - 0.6199 \cos (-26.5) = 2 - 0.55 = 1.44$$

$$ynew = yold - 0.6199 \sin (-26.5) = 1.5 + 0.276 = 1.77$$

Therefore, the midpoint at (2,1.5) has been moved to (1.44,1.77) or closer to Point J as desired.

To prevent the possibility of division by zero, before the value of SIGMA is computed, a check is made which determines if the absolute difference between the x-coordinates of the two points being considered is less than COMPAR (10^{-9}). If so, the value of SIGMA is set equal to 1.5707963268 radians ($\frac{\pi}{2}$), and the execution of the subroutine continues. In this fashion, the location of each midpoint is adjusted prior to the determination of the equation of its bisector.

Discussion of the Solution to the Simultaneous Equations. As previously mentioned, if the bisector is not vertical or horizontal, its endpoints are found from the simultaneous solution of the equations of the circumscribing circle and of the bisector. Since two endpoints are to be found (two solutions), the quadratic formula is utilized. The following definitions are used in the development of the solution to the simultaneous equations:

CIRCLE	BISECTOR UNDER CONSIDERATION
R = Radius of the circle	UMID = x-coordinate of midpoint
CX = x-coordinate of center of circle	VMID = y-coordinate of midpoint
CY = y-coordinate of center of circle	M2 = the negative inverse of the slope of the imaginary line connecting the two signal points
X = x-coordinate of point of intersection	
Y = y-coordinate of point of intersection	

The equation of the circle becomes:

$$R^2 = (X - CX)^2 + (Y - CY)^2 \tag{85}$$

Given a point Z, (X_z, Y_z) and a point J, (X_j, Y_j), the slope of the imaginary connecting line would be given by:

$$\text{Slope} = \frac{Y_j - Y_z}{X_j - X_z} \tag{86}$$

Therefore, the slope of the bisector becomes:

$$M2 = \frac{X_j - X_z}{Y_z - Y_j} \tag{87}$$

The point-slope form of the equation of a straight line is:

$$Y - Y_{11} = (\text{Slope})(X - X_{11}) \tag{88}$$

Substituting UMID and VMID for X_{11} and Y_{11} and M2 for the slope, the equation of the bisector becomes:

$$Y = (M2)(X - UMID) + VMID \tag{89}$$

The values of R, CX, CY, M2, UMID, and VMID are known, and the problem is reduced to that of two unknowns, X and Y, and two equations. The quadratic formula is used to solve for the value of one x-coordinate, and then its corresponding y-coordinate is determined. Then the other x-coordinate and its y-coordinate are computed. As a point of interest, the equations for the variables A, B, and C of the quadratic formula become:

$$A = 1.0 + (M2)^2 \tag{90}$$

$$B = (2)(M2)(VMID) - 2(CX) - (2)(M2)^2(UMID) - (2)(CY)(M2) \tag{91}$$

$$C = (CX)^2 + (CY)^2 - (R)^2 - (2)(M2)(UMID)(VMID) + (M2)^2(UMID)^2 + (2)(CY)(M2)(UMID) - (2)(CY)(VMID) + (VMID)^2 \tag{92}$$

Prior to solving for the x and y-coordinate values, however, a check is made to assure that the term under the radical in the quadratic formula, $B^2 - 4AC$, is positive. An error terminating the program execution would occur if this term were negative; hence, should this condition be found, an error message is provided and program returns to the beginning. As previously mentioned, once the coordinates of the endpoints of all the bisectors have been determined, they are stored in matrix IN.

Subroutine POINTS. The function of this subroutine is to determine the coordinates of the points of intersection of all the bisectors. Subroutine BISECT has computed the endpoints of all the bisectors and since any line can be described by two points, the equation of each bisector is known. This subroutine considers all possible combinations of two bisectors and the simultaneous solution of their equations provides the coordinates of the point of intersection. Several simplifying checks are performed first, however. For example, if either of the bisectors under consideration is vertical, then the y-coordinate of the point of intersection is known and inserting this into the equation of the nonvertical bisector easily provides the coordinates of the point of intersection. Several other simplifying conditions can exist and subroutine POINTS checks the following:

1. First bisector vertical?
2. Second bisector vertical?
3. First bisector horizontal?
4. Second bisector horizontal?
5. Bisectors parallel?

The reader is invited to analyze the subroutine flowchart to discover in what order these tests are made and how the simplification occurs. In fact, these tests are necessary in many instances (particularly when the bisectors are vertical) in order to prevent division by zero when the slopes are infinite. Before a computed point of intersection is stored in matrix IN, one final check is performed. If the bisectors are parallel, there is, of course, no point of intersection. It is also possible that the point of intersection may exist outside the circumscribing circle. Therefore, the final check is to assure that only points within the circle are stored, otherwise the initialized value "PHI" is retained to designate that no usable point of intersection exists. The use of subroutine SCALER would allow for the dimensions of the window (hence the circumference of the circle) to increase and thus allow those points previously outside the circle to be enclosed in the enlarged circle.

Subroutine REGION. This subroutine determines which of the points of intersection found in subroutine POINTS are actually the endpoints of the line segments which make up the decision boundary lines. Matrix IN has been constructed such that each time it is filled in for a particular lead point, it contains the coordinates of the points of

intersection of all the bisectors associated with that lead point. Since the signal probabilities are not necessarily equally likely, the concept of equal distance for determining the decision boundaries is not valid. The diagram below will be used to demonstrate the procedure used in subroutine [illegible]N. The s[illegible] [illegible] points A, B, C, and D. The six bisectors have been drawn in and labelled AB, AC, AD, BC, BD, and CD and the appropriate segments of the bisectors darkened to indicate the actual decision region boundaries.

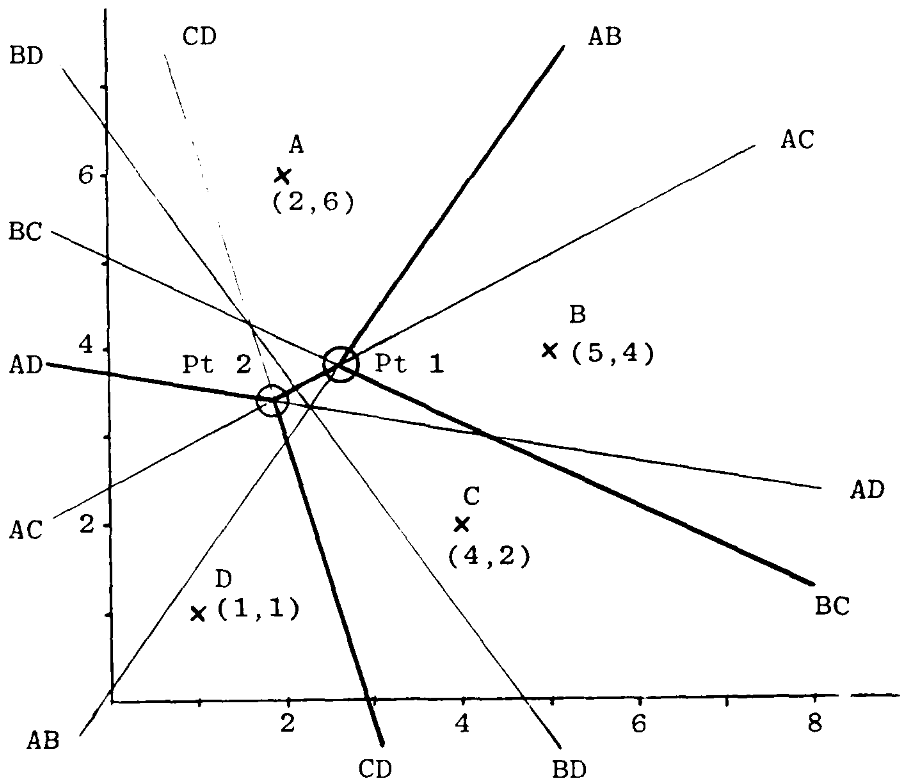


Fig 6. Bisectors and Decision Boundaries for Sample Signal Set

As is easily seen, each boundary segment separates only two signal points (unless one bisector should overlay another). The basic idea then is to pick a signal point (again called the lead point) and analyze all of its bisectors. The segment of any bisector which could be part of [illegible] that [illegible] clo-est to the lead point. Looking at Figure 6, we see that for signal "A", the portion of bisector "AB" above intersection Point 1 is part of the boundary. There is no other segment of any bisector of "A" which is closer to "A". However, for the portion of bisector "AB" below Point 1, bisector "AC" is closer, hence the decision boundary "bends" at this point of intersection. This subroutine thus checks each point of intersection on each bisector (in effect, checking each line segment) for the condition above.

The algorithm takes each of the bisectors in matrix IN one at a time and successively considers each point of intersection on that bisector. It then computes the slope of an imaginary line (called here a "connector") which connects the lead point and the point of intersection. Taking each of the other bisectors of that lead point in turn, it determines the point of intersection of the connector and this *next* bisector. (This point of intersection will be called point *new*). If point *new* is between the lead point and the original point of intersection, then the original point of intersection cannot be a boundary segment endpoint.

If point new is not between the lead point and the original point of intersection, then the next bisector is considered. If all the other bisectors are considered and no other bisector is found to be closer, then the original point of intersection is an endpoint of a decision boundary segment. The coordinates of this point of intersection are [illegible] a new matrix named "PPT". In this way all the points of intersection on each bisector are considered and PPT eventually contains the coordinates of all the endpoints of the decision boundary line segments.

The construction of matrix PPT is interesting as it is used by subroutine DECIDE to display the decision boundaries and subroutine EXACT to compute the probability of error for each signal. In order for subroutine DECIDE to draw a line segment, it needs to know the coordinates of each endpoint, or two coordinate pairs. This becomes the basic structure of matrix PPT. It is a 170 row by 5 column matrix. The first two columns always contain the x and y-coordinates of the starting point of a line segment, and columns three and four contain the x and y-coordinates of the end of that line segment. Column five is used to store the number of the lead point for that particular line segment. The 170 rows then allow for a total of 170 boundary line segments to be drawn. The very first boundary line start point for the first lead point is put in PPT(1,1) (x-coordinate) and PPT(1,2) (y-coordinate). The next

boundary line point is put in PPT(1,3) and PPT(1,4) as an endpoint for that segment and this same coordinate pair is also put in the first two columns of the next row of PPT, i.e., PPT(2,1) and PPT(2,2), so that this point becomes the starting point for the next line segment to be drawn. Con- [illegible] lead point are inserted in order into matrix PPT. This double placement of all remaining boundary line points for this lead point is performed until all its boundary line points are stored. The very first boundary line start point for the next lead point overwrites the last coordinate pair in columns one and two of PPT determined for the first lead point, since that coordinate pair is not needed as a start point. In the manner for the first lead point, all the boundary line points for this lead point are stored. After all the signal points have been the lead point, the operation is complete and control of the program returns to mainline.

Subroutine PERROR. The function of this subroutine is to use the concept of the Union Bound to compute an upper value for each signal's probability of error. It considers the probability of each signal, its location, and the system noise energy to determine this value. The presentation in Chapter II indicated that Union Bound on the system's total probability of error could be computed from Eq (74) which is provided again below:

$$P_e \le \sum_{j=1}^{K} P\{m_j\} \sum_{\substack{\ell=1 \\ \ell \ne j}}^{K} Q \left[\frac{d_{\ell j}}{2\sigma} - \frac{\sigma}{d_{\ell j}} \ln \left(\frac{P\{m_j\}}{P\{m_\ell\}} \right) \right] \tag{74}$$

From this it is possible to determine the probability of error for any signal in particular by writing the equation

$$P\{e|m_k\} \le \sum_{\substack{\ell=1 \\ \ell \ne k}}^{K} Q \left[\frac{d_{\ell k}}{\sqrt{2N_o}} - \frac{\sqrt{N_o/2}}{d_{\ell k}} \ln \left(\frac{P\{m_\ell\}}{P\{m_k\}} \right) \right] \tag{75}$$

where the substitution $\sigma = \sqrt{N_o/2}$ has been made.

Subroutine PERROR basically sums all the Q-functions for each signal point (or value of k). Subroutine BISECT has computed the distances between all the signal points and stored them in matrix DISTA. This matrix is used to obtain the values of $d_{\ell k}$. Prior to computing a particular Q-function value, a check is made to determine if $\ell = k$. If so, that particular Q-function is not computed and ℓ is incremented to perform the next computation. It is important to note that the IMSL Library subroutine used to compute the statistics of the Normal Distribution returns the probability that a random variable is less than some given value Y. The Q-function is the probability that a random variable is greater than that value of Y. Therefore, the Library-returned value is subtracted from unity to obtain the correct Q-function value. After all the Q-function values for a

particular lead point are computed, their sum becomes that signal's union bound on the probability of error and is stored in array "ERROR". After all errors have been computed, program control returns to mainline.

Subroutine EXACT. The function of this subroutine [illegible] the probability of error for each signal and the system as a whole. This more accurate probability is the result of actually integrating (to a close approximation) the decision area of each signal. To accomplish this, the subroutine uses matrix PPT to provide the coordinates of the endpoints of the line segments which make up the decision boundary regions. The *area* bounded by each signal is computed using the IMSL Library subroutine for the Bivariate Normal Distribution. This *area* is then subtracted from unity to provide the probability the signal falls outside the signal's boundaries, hence the probability of error.

The scale is enlarged to assure that all boundary intersections are within the circumscribed region. Then each signal is considered in turn and the entire signal set is translated so that the signal in question is located at the origin. This is done in order to normalize the resulting computation of the density function. Subroutine CMPUTE is called to provide the decision boundaries for the translated signal set. The rows of PPT which pertain to the signal in question are determined, and from these sets of coordinates

the maximum and minimum y-coordinate values are found. To compute the *area* of the signal's bounded region, the region is separated into numerous rectangles and the *areas* of these are computed and summed. The rectangles are determined as follows.

An imaginary horizontal line is positioned at the minimum y-coordinate value found above and successively moved up by an incremented amount. As the line is moved up, the x-coordinate values of the points of intersection of this horizontal line and the boundary segments are determined (see Figure 7). These x-coordinates become the X-limits of integration and the y-coordinates of integration are the value of the horizontal line and the next (incremented) position of the horizontal line. If there is only one boundary line (e.g., the other limit is at infinity), the *area* is integrated out to five standard deviations (STD).

For each small rectangular area computed, four actual integrations are required. This is necessary because the IMSL routine used, MDBNOR, computes the Bivariate Normal Distribution only from negative infinity in both dimensions to the point (X,Y). Hence, since the lower limits cannot be specified, there are *areas* included in the computation which must be subtracted off. The figure shows a sample rectangular region in red with single hash marks. In order to determine the *area* of this rectangle, first the *area* of everything below and left is computed using the bounds

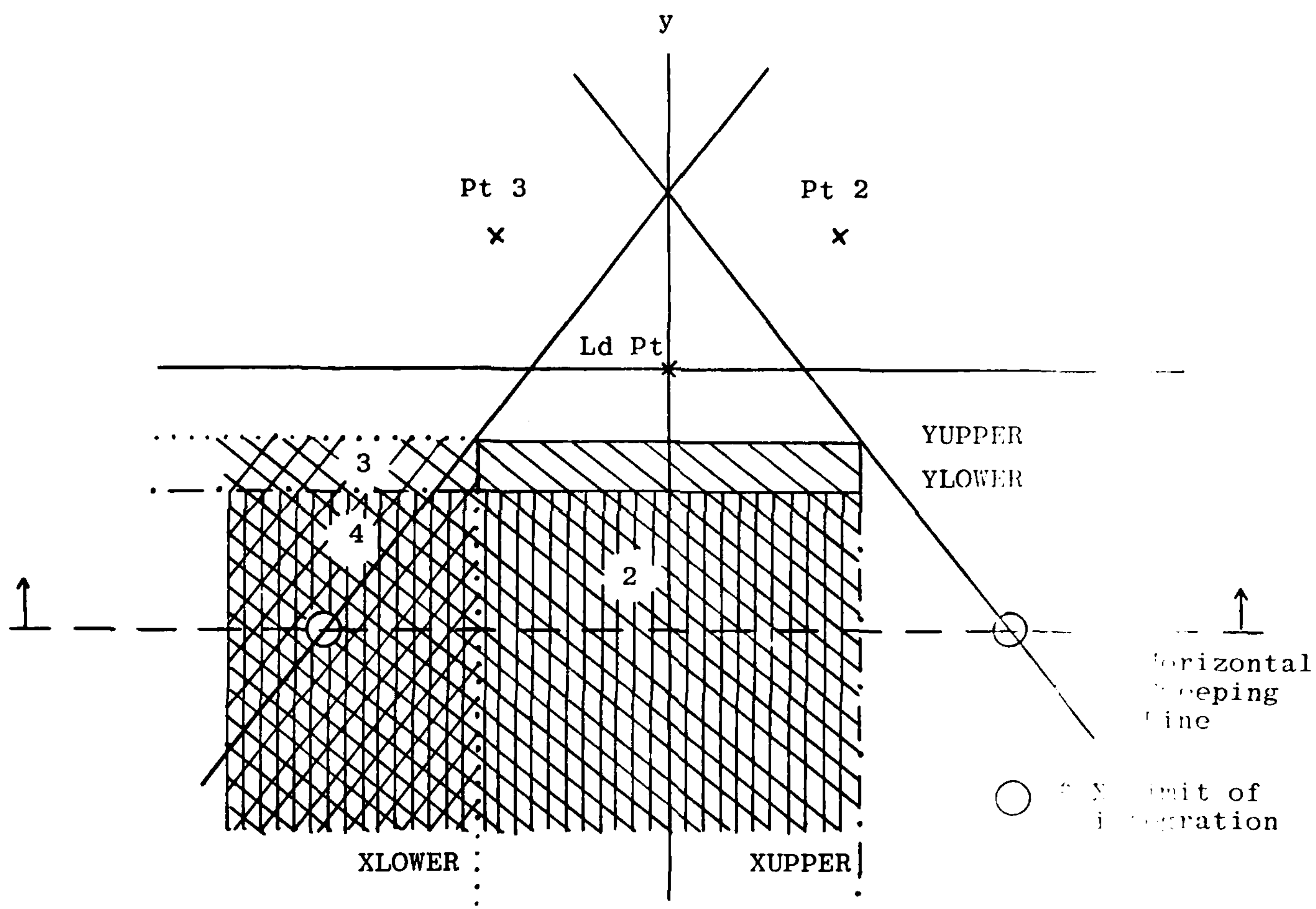


Fig 7. Computation of Probability Density of Rectangular Region

XUPPER and YUPPER as the limits of integration. Secondly, from this total *area*, both *areas* of double [illegible]ash marks must be subtracted. Hence the *area* bounded by XUPPER and YLOWER (Area 2 with boundary —·—·), and the *area* bounded by XLOWER and YUPPER (Area 3 with boundary ······), are com- [illegible] subtracted. Th[illegible] is[illegible] the [illegible] lower left portion bounded by XLOWER and YLOWER (Area 4) to be subtracted out twice. Therefore, it too is computed and then added back. Thus, the *area* of the rectangle is given by:

$$\text{Area of Rectangle} = \text{Total Area} - \text{Area 2} - \text{Area 3} + \text{Area 4}$$

This result then is stored and after all the rectangular regions for the signal in question summed, the total is subtracted from unity to yield the probability of error for that signal.

The details of the operation of this routine will be discussed as follows:

1. Determination of the limits of integration
2. Determination of variable incrementing factor
3. Adjustment of limits when noise variance is not equal to one

In order to gain an understanding of the algorithm used to determine the limits of integration, a simple example will be discussed. The signal set to be used is

given by:

$$s_1 = \begin{bmatrix} 2 \\ 0 \end{bmatrix} \qquad s_2 = \begin{bmatrix} 0 \\ 2 \end{bmatrix} \qquad s_3 = \begin{bmatrix} -2 \\ 0 \end{bmatrix} \qquad s_4 = \begin{bmatrix} 0 \\ -2 \end{bmatrix}$$

which is plotted as

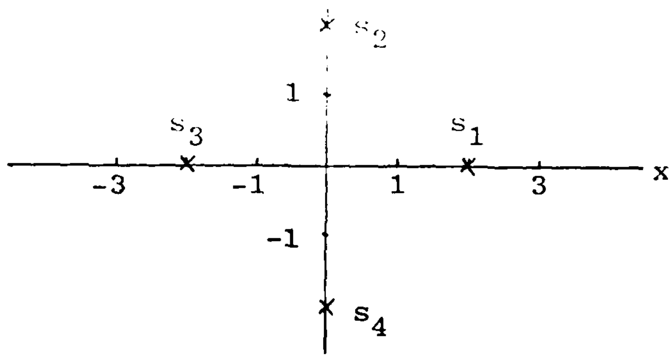


Fig 8 Sample Signal Set for Explanation of Subroutine EXACT

For the example of the computation to be performed, the probability of error for Signal s_2 will be described. It will be assumed that the window has been properly scaled. As the overview outlines, the signal set is translated so that Signal 2 is at the origin and subroutine CMPUTE is called to determine the boundary line segments and their endpoints. If the signal set were plotted and the decision boundaries drawn in at this point, the display would be shown as in Figure 9 with the top of the window at least five STD above s_2.

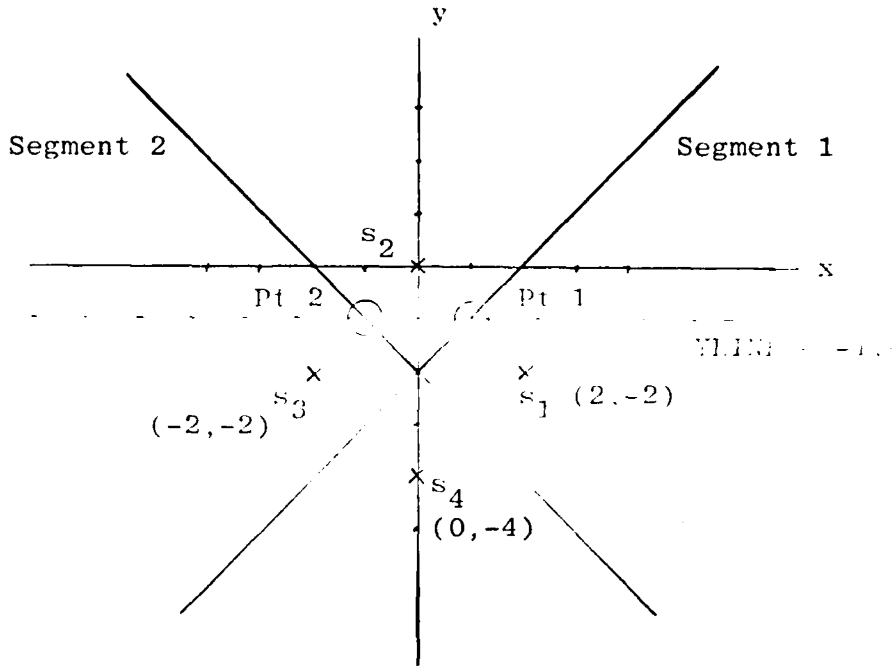


Fig 9. Translated Sample Signal Set

Since column five of matrix PPT identifies to which signal point each row belongs, the rows associated with Signal 2 are easily determined. From these rows, the minimum and maximum y-coordinate values are found. The minimum value will become the starting location of the horizontal line (YLINE) which will sweep up over the entire area bounding Signal 2. The maximum value (YTOP) will become the ultimate limit of the sweep of the YLINE. With YLINE at its starting position, the subroutine enters the top of several loops to determine the values of the x-coordinate limits. (The first time through for this example, YLINE is at y = -2.0 and the X-limits are identical so the *area* equals zero and YLINE is

incremented.) For the purposes of explanation, it will be assumed that YLINE has been incremented to y = -1.0 and the procedure is joined in progress. The routine considers the first boundary segment of Signal 2 (that segment from [0, -2] to the upper right corner) and determines the intersection [illegible] point of intersection lies numerically between the values of the x-coordinates of the endpoints of the segment and YLINE lies numerically between the values of the y-coordinates of the endpoints of the segment, then this point of intersection will be a limit of integration. This is Point 1 in the figure. The routine will then consider the next boundary line segment (i.e., the next row of PPT) and determine its intersection with YLINE. It finds it to be Point 2 in the figure. The maximum of the two limits becomes the X-limit of integration identified as XUPPER and the minimum becomes the X-limit named XLOWER. These two values are then stored respectively as XUPOLD and XLOOLD. YLINE is incremented and the next two values for the X-limits computed. These two values, named XUPNEW and XLONEW are compared with the two previous quantities. The lesser of XUPNEW and XUPOLD becomes the value of XUPPER and the greater of XLONEW and XLOOLD becomes the value of XLOWER used in the integration. This assures that the integrated probability of a correct decision as computed is slightly less than the actual probability. Thus, the probability of error that results

will always be slightly greater than the actual probability of error and we are assured of an upper bound. The current value of YLINE becomes the value of YUPPER and decrementing this quantity yields the value of YLOWER. With these four values determined, the integration process described in the [illegible] ried out and [illegible] routine starts again with the top of the "X-limit" determining loops. For the example under consideration, YLINE will continue to sweep up the Signal 2 region stopping at each increment to have new X-limits determined and more and more rectangular *areas* integrated. It will stop when it reaches a value equal to either YTOP, which would mean that the top of the bounded region has been reached, or when it is equal to the value of five STD. (Five STD is set as an automatic limit because negligible *area* exists any further up.)

The reader will note that several other configurations for the decision region boundaries are possible. Each of these is dealt with at an appropriate location in the subroutine and the interested reader is urged to follow the flowchart for the actual *implementation*. However, the possibilities will be identified below and the actions taken by the routine described.

The first condition which may occur is that all boundary segments may be checked using a particular value of YLINE and no X-intersections are found. This implies the signal under consideration has only one boundary segment

and it, in fact, is horizontal. Since the values of YLINE and YTOP will be equal (that is, equal to the y-coordinate value of the boundary segment), the starting value for YLINE and the value for YTOP must be assigned automatically. In order to do this, a check is made to determine if the bound- [illegible] boundary is above the signal point. YLINE is set equal to a negative five STD and YTOP is set equal to the lesser of the y-coordinate value of the boundary or a positive five STD. Conversely, if the boundary is below the signal point, YLINE is set equal to the greater of the y-coordinate value of the boundary or a negative five STD and YTOP becomes equal to positive five STD. Since there are no computed limits on the values of X, the routine automatically assigns XLOWER equal to a negative five STD and XUPPER equal to a positive five STD.

A second special condition occurs when all the boundary segments are checked using a particular value of YLINE and only one X-intersection is determined. This again implies that the signal point has only one boundary segment, but that the boundary this time is inclined. In this instance, another check is required. This additional check (named XCHECK) determines on which side of the signal point the boundary line rests. If the boundary is to the left of the signal point, XLOWER becomes the value of the x-coordinate of the point of intersection and XUPPER is automatically

AD-A115 498 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOO--ETC F/G 9/2
ANALYSIS OF THE OPTIMUM RECEIVER DESIGN PROBLEM USING INTERACTI--ETC(U)
DEC 81 M R MAZZUCCHI

UNCLASSIFIED AFIT/GE/EE/81D-39 NL

2 OF 3

AD
A115498

assigned to be a positive five STD. If the boundary is to the right of the signal point, the opposite situation occurs, and XLOWER is automatically set to a negative five STD.

Due to the nature of the construction of matrix PPT, [illegible] li[illegible] segm[illegible] once (particularly if any bisectors happen to overlay one another). Additionally, a particular boundary line segment may be duplicated but with its endpoints reversed. For these reasons, after the first X-limit is found, every future boundary line segment considered is first checked for either of the conditions above. Unless the segment is distinct from the first usable boundary segment, it will be rejected and the search will continue.

This, then, is the embodiment of subroutine EXACT. Regardless of the shape of the bounded region, a starting location for YLINE is determined and it step by step works its way up to and over the signal point in question, creating rectangular regions which, when summed, equal the probability of a correct decision being made. This value subtracted from unity results in the probability of error.

The second specific operation of subroutine EXACT to be discussed is the determination of the variable incrementing factor. Due to the nature of the Bivariate Normal Distribution Function, the use of a constant value to increment YLINE is inefficient. This is because the further one moves

from the signal, the smaller the value of the function, and hence the smaller the computed *area* of a rectangular region. Therefore, when YLINE is at the extremes of its movement, the value of the increment must be its largest; as YLINE approaches the signal, the incrementing value must be made smaller. This results in a closer approximation to the actual value of the function within the bounded region. Note the simplified diagram below:

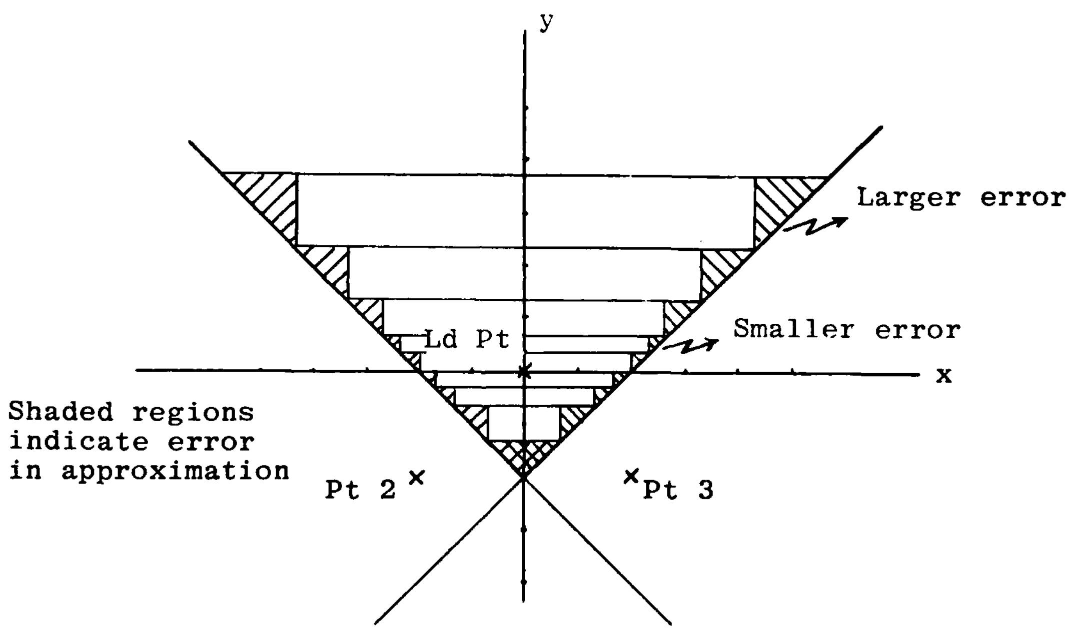


Fig 10. Diagram of Variable Incrementing Value

There are three critical variables which must be considered when designing this "sliding" computation. First, what is the initial incrementing value to be? Second, what criterion should be used to initiate a change of value? And finally, how much should this value be reduced and then

enlarged? The answers to all three of these questions could become a thesis in their own right. The numerical analysis required for the most efficient combination for all possible signal sets would be enormous. However, one can use the Tables of the Circular Bivariate Normal Distribution, experiment with the routine, and much trial and error and arrive at a relatively acceptable set of answers. The results used in the subroutine are as follows:

1. In the interest of using the smallest amount of CPU time and yet still striving for as close an approximation as possible, the initial incrementing value is equivalent to one one-hundredth (0.01) of the total length traversed by YLINE (i.e., YTOP - YBOT).

2. The criterion used to initiate a change of the incrementing value is the value of the function returned. As YLINE nears the signal point, the value of the function or the *area* of the rectangles increases, indicating a need to take more samples or decrease the incrementing value. A problem arises here, for if the incrementing value is decreased too much or too quickly or both, the value of the function will again be very low, thus signaling a need to prematurely increase the incrementing value. This would cause an oscillation of the incrementing value which is clearly undesirable and leads to greater inaccuracies than using a simple fixed incrementing value. Since the accuracy of the value of the function returned by the IMSL subroutine

used to compute the density is given as 0.00001, this is the value used to signal the first decrease in the incrementing value. The next level used is ten times this, or 0.0001, and the last step occurs when the returned value of the function is 100 times greater than the initial value or 0.001.

3. The final consideration is the amount the incrementing value should be decreased. It was found that simply halving the value at each step appeared to provide adequate tracking.

Lastly, in order to provide the capability to obtain an even closer approximation (hence greater resolution), any or all of these factors could be made to vary according to the user's desires. It became a design decision to allow only the first variable to be user altered and even this is still not solely his choice. The closer approximation is computed by allowing the integrations to be more closely spaced. This is accomplished by allowing the user to successively halve the initial incrementing value, perform the operation, and check the results. This procedure can be continued indefinitely (assuming one has unlimited CPU time). Hence, the user is allowed to obtain the more accurate answer, but not at the expense of attempting to "reinvent the wheel."

The third special operation of subroutine EXACT, which will be explained, is the adjustment of the limits of

integration. For the reasons discussed below, these limits must be adjusted whenever the noise power spectral density or variance (i.e., variance of the probability distribution) is not unity.

The Bivariate Normal Distribution Function is given by:

$$P(x,y) = \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-\rho^2}} \exp\left\{ -\frac{1}{2(1-\rho^2)} \left[\left(\frac{x-\mu_x}{\sigma_x}\right)^2 - 2\rho \frac{(x-\mu_x)(y-\mu_y)}{\sigma_x\sigma_y} + \left(\frac{y-\mu_y}{\sigma_y}\right)^2 \right] \right\} \tag{93}$$

For this project, the assumptions allow the following simplifications:

$$\rho = 0 \quad ; \quad \mu_x = \mu_y = 0 \quad ; \quad \sigma_x = \sigma_y = \sigma$$

which result in the following representation for this function:

$$P(x,y) = \frac{1}{2\pi\sigma^2} \exp\left\{ -\frac{1}{2} \left[\frac{x^2}{\sigma^2} + \frac{y^2}{\sigma^2} \right] \right\} \tag{94}$$

The IMSL subroutine MDBNOR, which computes the Bivariate Normal, additionally assumes that the variance equals one, $\sigma^2 = 1.0$. It, therefore, computes the following density:

$$p(x,y) = \frac{1}{2\pi} \int_{-\infty}^{x} \int_{-\infty}^{y} \exp\left\{-\frac{(x^2 + y^2)}{2}\right\} dx\, dy \tag{95}$$

Hence, in order to use it with a variable variance, i.e., other than just unity, the limits on X and Y must be manipulated. The density we desire to compute is given by:

$$p(x,y) = \frac{1}{2\pi\sigma^2} \int_{-\infty}^{x} \int_{-\infty}^{y} \exp\left\{-\frac{(x^2 + y^2)}{2\sigma^2}\right\} dx\, dy \tag{96}$$

which (since the correlation coefficient, ρ , is zero) can be written:

$$p(x,y) = \frac{1}{2\pi\sigma^2} \int_{-\infty}^{x} \int_{-\infty}^{y} \exp\left[-\frac{x^2}{2\sigma^2}\right] dx \cdot \exp\left[-\frac{y^2}{2\sigma^2}\right] dy \tag{97}$$

Using the change of variables method we allow:

$$\frac{v^2}{2} = \frac{x^2}{2\sigma^2} \quad \text{and} \quad \frac{w^2}{2} = \frac{y^2}{2\sigma^2}$$

$$v = \frac{x}{\sigma} \quad \text{and} \quad w = \frac{y}{\sigma}$$

$$dv = \frac{1}{\sigma} dx \quad \text{and} \quad dw = \frac{1}{\sigma} dy$$

$$dx = \sigma dv \quad \text{and} \quad dy = \sigma dw$$

By substitution into Eq (97) above, we have:

$$p(v,w) = \frac{1}{2\pi\sigma^2} \int_{-\infty}^{v} \int_{-\infty}^{w} \exp\left[-\left(\frac{v^2}{2}\right)\right] \sigma dv \cdot \exp\left[-\left(\frac{w^2}{2}\right)\right] \sigma dw \qquad (98)$$

Simplifying and adjusting the limits results in:

$$p(v,w) = \frac{1}{2\pi} \int_{-\infty}^{\frac{x}{\sigma}} \int_{-\infty}^{\frac{y}{\sigma}} \exp\left[-\left(\frac{v^2}{2}\right)\right] dv \cdot \exp\left[-\left(\frac{w^2}{2}\right)\right] dw \qquad (99)$$

This is the desired form and implies that the only "correcting factor" necessary is an adjustment to the limits of integration. That is, each limit must be multiplied by $\frac{1}{\sigma}$ in order to correct for the noise variance equalling something other than unity. Subroutine EXACT uses the substitution, $T = \frac{1}{\sigma}$, to simplify the expression.

<u>Subroutine CMPUTE</u>. This subroutine forms the basis for the computation and display of the decision regions. Given the particular signal set and the scaling factor, it calls subroutine WINDOW to calculate the dimensions of the window and the equation of the circumscribing circle. Then using successive calls to subroutines BISECT, POINTS, and REGION, it determines which points of the bisector intersections are the endpoints of the segments which make up the decision boundary regions.

The initial call to subroutine WINDOW is performed only once to calculate the window dimensions (XMIN, XMAX, YMIN, YMAX), and the radius of the circle. Then, for each signal point, each of the other three subroutines is called in order. Subroutine BISECT determines the equations of all the bisectors for that point. Subroutine POINT then finds all the points of intersection for these bisectors and subroutine REGION determines which of these points of intersection should be connected to display the decision region boundaries. Matrix IN is used to store the endpoints (hence the equations) of the bisectors and all the points of intersection. Matrix PPT is used to store the coordinates of the segment endpoints. Hence, every time CMPUTE is called, the entire signal set is considered and all the decision regions determined.

Output Subroutines

Now the program is ready to prompt for the user options. The signal set can be altered, the system parameters changed, or the statistics and boundary regions for the current signal set displayed. In order to do this, one of the output subroutines must be called. Table III lists the output subroutines along with the functions they perform.

TABLE III

OUTPUT SUBROUTINES

Subroutine Name	Function
PLOT	Provides graphical display of signal set and listing of coordinate pairs
DECIDE	Provides graphical display of signal s[illegible]
OUTPUT	Provides tabular listing of signal and system statistics
EXACT	Provides the actual probability of error it has computed

Subroutine PLOT. The purpose of this subroutine is to allow the user to visually display and check the signal set currently under consideration. In addition to a plot of the signals, the routine can also provide a listing of the signal set coordinate pairs. Finally, it serves as the display for subroutine ADDGRA, which allows the user to graphically add signal points. The routine begins by clearing the screen and displaying the X-Y axis and the window boundary. It then considers each signal point in turn, moves the cursor to the point location, and then does a small relative movement to the left and down the screen in order to center the point marker, an asterisk, exactly on the signal point location. Another small relative movement is made and the number (or letter, for signals greater than nine) of the signal is displayed. After all the signals have been positioned, the cursor is moved below the display and a prompt

provided the user. Since he may not be interested in a listing of the coordinate pairs, he is given the choice of clearing the screen and having the listing provided or leaving the subroutine at this point to continue program execution. This prompt also serves as a pause to allow the [illegible]

Subroutine DECIDE. Subroutine DECIDE provides a graphical display of the signal set with the computed decision boundaries displayed. As previously discussed, the subroutines BISECT, POINTS, and REGION fill in and alter matrices IN and PPT. The contents of PPT are the endpoints of all the segments of the decision region boundaries. Matrix PPT has five columns and up to 170 rows. Columns one and two contain the x and y coordinates of one endpoint and columns three and four contain the x and y coordinates of the other endpoint of each segment to be drawn. Column five contains the number of the signal point to which this boundary segment belongs (see Subroutine REGION). The cells of PPT are initialized to contain PHI and are changed as the endpoints are filled in. This routine checks columns one and three of each row to assure a valid endpoint, i.e., the value is less than PHI. If both values are valid, a visible line connecting the coordinate pairs is drawn. Otherwise, the next row of PPT is checked and so on until all 170 rows have been considered. The routine then displays the signal

set points, draws the window boundaries, and draws in dashed lines representing the X-Y axis.

Subroutine OUTPUT. The function of this subroutine is to provide a tabular listing of all computed statistics except the integrated probability of error determined by subroutine EX[illegible] [illegible] the [illegible] subroutine EXACT can use considerable CPU time, it is relegated to a separate option for use with those signal sets appearing most promising. Subroutine OUTPUT then gives the user an expedient analysis of the signal set at hand. Specifically, for each individual signal point, the following data is displayed:

1. Signal number
2. Coordinate pair of signal
3. Signal probability
4. Signal energy
5. Signal-to-noise ratio
6. Union bound on the probability of error

Additionally, the system statistics listed below are provided:

1. System noise power spectral density (σ^2)
2. Total system signal energy
3. Total system probability of error (Union Bound)
4. Coordinates of the system center of gravity

Due to the range of values possible, an E format, or scientific notation format, is used in printing out the values of coordinate pairs and probabilities of error. For the

values of signal probabilities, their energies, and SNR, a decimal format is used.

Discussion of Matrix IN

Matrix IN (for intersection), is a large matrix (34 x 64) which is an integral element of subroutine BISECT, POINTS, and REGION. An understanding of its construction is essential to an understanding of these routines. Basically, it is a matrix which contains all the coordinate pairs of the points of intersection of the perpendicular *bisectors* and a circle which circumscribes the signal set and the *bisectors*. The word *bisectors* is italicized because the lines will only be bisectors when the probabilities of the two signals it separates are equal. However, the term will continue to be used with the understanding that the location of the perpendicular boundary line will be shifted toward the signal having the smaller probability of occurrence when the probabilities are not equal.

Subroutine BISECT determines the coordinates of intersection of each bisector and an imaginary circle which circumscribes the signal set. The first row of IN contains the x and y-coordinates of one end of each bisector. The second row of IN contains the x and y-coordinates of the other end of each bisector.

Subroutine POINTS uses these endpoints to determine the equations of all the bisectors. It then determines all

the coordinates of all the points of intersection of the bisectors which fill out the rest of IN.

Mapped out, Figure 11 shows how the points of interest are inserted in IN for a four point signal set. The number of bisectors between Point 1 and the other three sig- [illegible] n-1. [illegible] becomes (n-1) x 2 to allow for the x and y-coordinates of each endpoint. The number of rows is (n-1) + 2 because in addition to the bisector points of intersection, the endpoints of each bisector are in rows one and two. The numbering of the bisectors is demonstrated in the figure, and shows that the lead point is fixed and the second point is incremented until the entire signal set is considered. After all the operations and computations for the first lead point have been accomplished, the matrix is reinitialized and used for the next lead point. In this fashion, all the signal points are processed.

	Bisector Between Points 1 & 2		Bisector Between Points 1 & 3		Bisector Between Points 1 & 4		
	x-coordinate	y-coordinate	x-coordinate	y-coordinate	x-coordinate	y-coordinate	....
Top of Circle							
Bottom of Circle							
Bisector 1,2							
Bisector 1,3							
Bisector 1,4							

Fig 11. Organization of Matrix IN

IV. Program Verification

Introduction

In order to completely verify this program, one would be forced to test every conceivable combination of [illegible] task [illegible] settling for a test of only the extreme values of each variable, but even this would be an enormous undertaking. However, as a tutorial tool, rather than a commercial product, the primary interest should be how well the program performs with a set of "typical inputs." It is essential that it be able to properly process the kinds of operations or signal set manipulations which a curious student may be inclined to investigate.

For these reasons, the program verification consists of the comparison of solutions for a tutorial problem. The statement of the problem is given, the hand calculated solutions explained, and then the program generated results presented and compared. Part one of the verification begins with a description of the initial problem signal set and a comparison of the results obtained by the two methods. Next, the system noise energy will be varied and the two solutions again compared. Returning to the original problem, the individual signal probabilities will be varied and the solutions analyzed. The final set of comparisons is for the results when a signal of the original set is deleted and

a new signal added. In this manner, the majority of the mathematical aspects of the program will be tested.

A major objective of the project is the selection of an optimum signal set. Since the criterion of optimization chosen is the probability of error criterion, part two of [illegible] Two important aspects of this computation are shown. First, the amount of deviation in the calculated value as the signal set is rotated is presented and, second, the fact that the calculated value is always an upper bound is verified.

Part three tests a third essential purpose of the program: the graphical display capability. The verification of the graphics consists of the display of a sample signal set and demonstrates the results of translation, rotation, and scaling of this signal set.

Manipulation of Signal and System Parameters

Sample Problem and Initial Comparison. In a quadriphase communication system, the four signal vectors are given as

$$s_1 = \begin{bmatrix} 2 \\ 2 \end{bmatrix} \quad s_2 = \begin{bmatrix} -2 \\ 2 \end{bmatrix} \quad s_3 = \begin{bmatrix} -2 \\ -2 \end{bmatrix} \quad s_4 = \begin{bmatrix} 2 \\ -2 \end{bmatrix}$$

Figure 12 is a graphical representation of this signal set. Initially, the signals are equiprobable and the noise variance is unity. The statistics we wish to compute and compare are the signal energies, signal-to-noise ratios, the

union bound on the probability of error, and the actual probability of error. From the symmetry of the given signal set, it is obvious that the statistics of each signal will be the same. Therefore, only one signal need be considered. The Handbook of Tables for Mathematics is used to determine [illegible] 9:929.929).

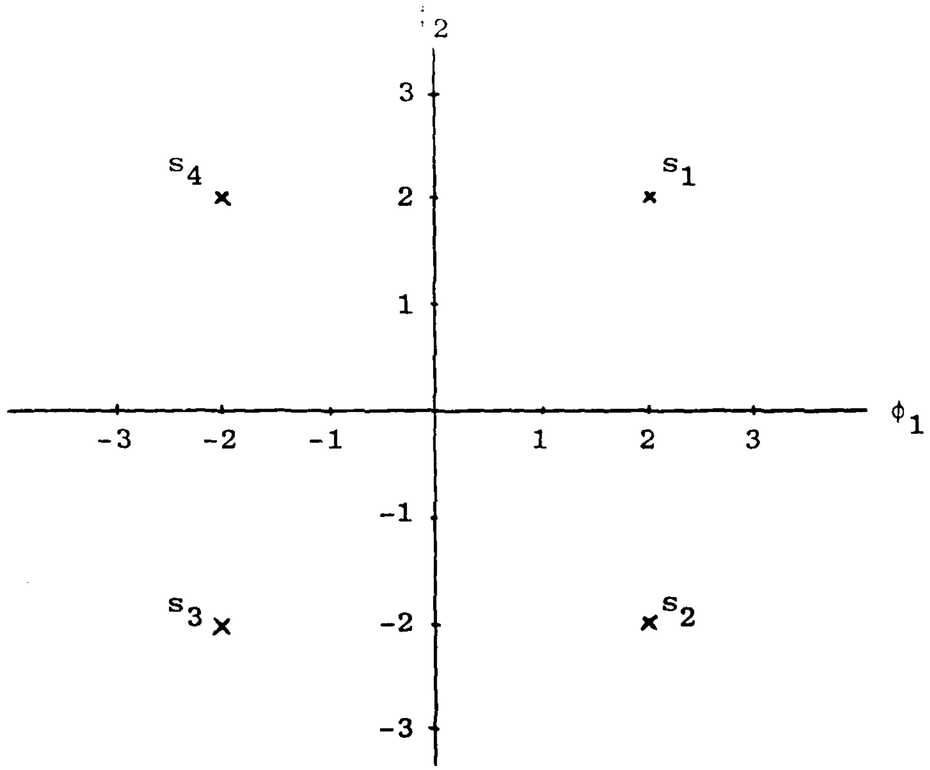


Fig 12. Quadriphase Signal Set Used For Verification

1. Signal Energy. From Eq (79), we recall the energy of a signal is computed as follows:

$$\text{Energy} = (\text{x-coordinate})^2 + (\text{y-coordinate})^2 \qquad (79)$$

Therefore, if the signal vectors are measured in volts across a resistance of one ohm, their energy is:

[illegible]

2. Signal-to-Noise Ratio. From Eq (78), the signal-to-noise ratio (SNR) is defined as:

$$\frac{E_s}{N_o} \Big|\, db \triangleq 10 \log_{10} \frac{E_s}{N_o} \qquad (78)$$

which gives a SNR for each signal of

$$\frac{E_s}{N_o} \Big|\, db = 10 \log_{10} \frac{8}{2} = 6.0206 \text{ db}$$

3. Union Bound on the Probability of Error. To determine the union bound, we must first compute the distances between every combination of two signals. Either by direct computation from Eq (65) or by inspection of Figure 12, it is easy to establish that the distances are as given in Table IV.

TABLE IV

Distances of Signal Set Used for Verification

	ℓ			
k	1	2	3	4
1	–	4	$4\sqrt{2}$	4
2	4	–	4	$4\sqrt{2}$
3	$4\sqrt{2}$	4	–	4
4	4	$4\sqrt{2}$	4	–

Using these in Eq (66), the union bound on the P_e for each signal becomes

$$P\{e|m_k\} \leq \sum_{\substack{\ell=1 \\ \ell \neq k}}^{K} Q\left[\frac{d_{\ell k}}{2\sigma}\right] \tag{66}$$

$$P\{e|m_k\} \leq Q(2) + Q(2\sqrt{2}) + Q(2) = 0.0479$$

4. For this particular signal set, it is possible to compute the exact value of the probability of error. For example, if message m_1 is sent, no error will be made unless n_1 or n_2 is greater than 2.0. Therefore, it can be shown (Ref 3:118), that the conditional probability of error is given by

$$P\{e|m_1\} = 2Q(2) - Q^2(2) = 0.0449$$

Figure 12 also shows the decision region boundaries for this signal set. Due to symmetry, the ϕ_1 and ϕ_2 axes actually

represent these boundary lines. Table V provides the program computed results for this problem. The data listed under OPTION> 9 for signal energy, SNR, and the union bound are nearly identical, as any differences are attributable to round-off error. OPTION> 12 provides the program calcu- [illegible] this particular (symmetric) signal set, the two computations are identical.

Variation One: Altering Noise Energy. The signal set for this example remains the same; however, in this case we vary the system noise energy. From the default value of 2.0, we choose to increase it to 8.0 and examine the effects on the system statistics.

1. Signal Energy. Since the signals have not been altered, no change in their energy is possible.

2. Signal-to-Noise Ratio. Referring to Eq (78) once again, the adjusted SNRs become

$$\frac{E_s}{N_o} \Big| \text{db} = 10 \log_{10} \frac{8}{8} = 0.0 \text{ db}$$

3. Union Bound on the Probability of Error. Since the signal set has not been altered, the distances in Table IV are still valid. A noise energy of 8.0 yields a noise variance, σ^2, of 4.0; hence, the value of σ used in Eq (66) must be 2.0. The union bound given by this equation thus becomes

TABLE V

Statistics of Original Signal Set Used for Verification

OPTION >9

SIGNAL	COORDINATES		PROBABILITY	ENERGY	SNR	UNION BOUND PROB OF ERROR
1	.200E+01	.200E+01	.2500	.800E+01	6.0206	[illegible]
2	-.200E+01	.200E+01	.2500	.800E+01	6.0206	[illegible]
3	-.200E+01	-.200E+01	.2500	.800E+01	6.0206	[illegible]
4	.200E+01	-.200E+01	.2500	.800E+01	6.0206	[illegible]

SYSTEM NOISE PSD : 1.

TOTAL SYSTEM ENERGY : .320E+02

TOTAL SYSTEM PROBABILITY OF ERROR: < .4784E-01

SYSTEM CENTER OF GRAVITY LOCATED AT 0. 0.

OPTION >12

SIGNAL	PROBABILITY	INTEGRATED PROB OF ERROR
1	.2500	.4498E-01
2	.2500	.4498E-01
3	.2500	.4498E-01
4	.2500	.4498E-01

TOTAL SYSTEM PROBABILITY OF ERROR: < .4498E-01

$$P\{e|m_k\} \leq 2Q(1) + Q(\sqrt{2}) = 0.3964$$

4. The exact probability of error is found in a manner similar to the previous example. As described in Chapter III, Eq (99), the limits of integration must be multiplied by $\frac{1}{\sigma}$; therefore, if message m_1 is sent, no error will be made unless n_1 or n_2 is greater than $2(\frac{1}{\sigma})$, or 1.0. Hence, the exact probability of error becomes

$$P\{e|m_1\} = 2Q(1) - Q^2(1) = 0.2922$$

The program calculated results for this problem are located in Table VI. OPTION> 9 contains the signal and system statistics for comparison. Again, the agreement with the manually obtained results is excellent. As before, OPTION> 12 gives the integrated value of P_e, and since the signal set is still symmetric, the two results are nearly identical.

Variation Two: Altering Signal Probabilities. For this particular variation, only the signal probabilities will change. The system noise variance is restored to unity and the signal set remains the same. The probabilities we will consider are

$$P\{s_1\}=0.01 \quad P\{s_2\}= 0.04 \quad P\{s_3\}= 0.2 \quad P\{s_4\}=0.75$$

1. Signal Energy. No change in signal energy has been made.

TABLE VI

Statistics of Variation One

OPTION IV

SIGNAL	COORDINATES		PROBABILITY	ENERGY	SNR	UNION BOUND PROB OF ERROR
1	.200E+01	.200E+01	.2500	.800E+01	0.0000	.3960E+00
2	-.200E+01	.200E+01	.2500	.800E+01	0.0000	.3960E+00
3	-.200E+01	-.200E+01	.2500	.800E+01	0.0000	.3960E+00
4	.200E+01	-.200E+01	.2500	.800E+01	0.0000	.3960E+00

SYSTEM NOISE PSD : 4.

TOTAL SYSTEM ENERGY : .320E+02

TOTAL SYSTEM PROBABILITY OF ERROR: < .3960E+00

SYSTEM CENTER OF GRAVITY LOCATED AT 0. 0.

OPTION SIX

SIGNAL	PROBABILITY	INTEGRATED PROB OF ERROR
1	.2500	.2921E+00
2	.2500	.2921E+00
3	.2500	.2921E+00
4	.2500	.2921E+00

TOTAL SYSTEM PROBABILITY OF ERROR: < .2921E+00

2. Signal-to-Noise Ratio. Because the original value of the system noise energy is used, the SNR will return to the initially computed value of 6.0206 db.

3. Union Bound on the Probability of Error. Since the signal probabilities are no longer equally likely, Eq (66) cannot be used. Instead, we refer to Eq (75) given below:

$$P\{e|m_k\} \leq \sum_{\substack{\ell=1 \\ \ell \neq k}}^{K} Q\left[\frac{d_{\ell k}}{2\sigma} - \frac{\sigma}{d_{\ell k}} \ln\left(\frac{P\{m_\ell\}}{P\{m_k\}}\right)\right] \tag{75}$$

Note that the distances of Table IV remain valid and we can proceed with the calculation for signal s_1. By substitution of the parameters into Eq (75), we have $P\{e|m_1\} \leq 0.2390$. In a similar fashion, the remaining values are computed and are given below:

$$P\{e|m_2\} = 0.0744$$
$$P\{e|m_3\} = 0.0561$$
$$P\{e|m_4\} = 0.0113$$

4. The computation of the exact P_e becomes more difficult in this variation. The signals are no longer equiprobable and the perpendicular bisectors which represent the decision region boundaries are shifted by an amount DELTA as demonstrated in Chapter III and Eq (83). Figure 13 shows the signal set with the decision region boundaries as they

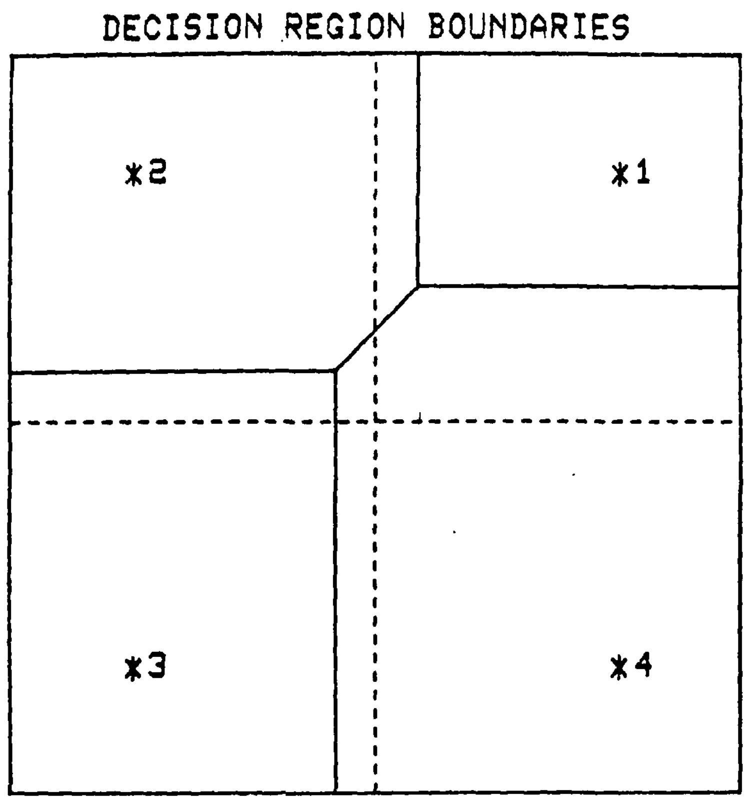


Fig 13. Decision Region Boundaries of
Variation Two

now appear. From this it is evident that the method previously used to calculate the exact P_e is legitimate only for signals one and three. Also in order to use that method for s_1 and s_3, we need to know how much the boundaries have been shifted. From Figure 14, we note that to determine P_e for these signals, we must calculate the values of Δ_{12}, Δ_{14}, Δ_{32}, and Δ_{34}. Equation (83) is provided again below:

$$\Delta_{ij} = \frac{N_o}{2d} \ln \frac{P\{m_i\}}{P\{m_j\}} \tag{83}$$

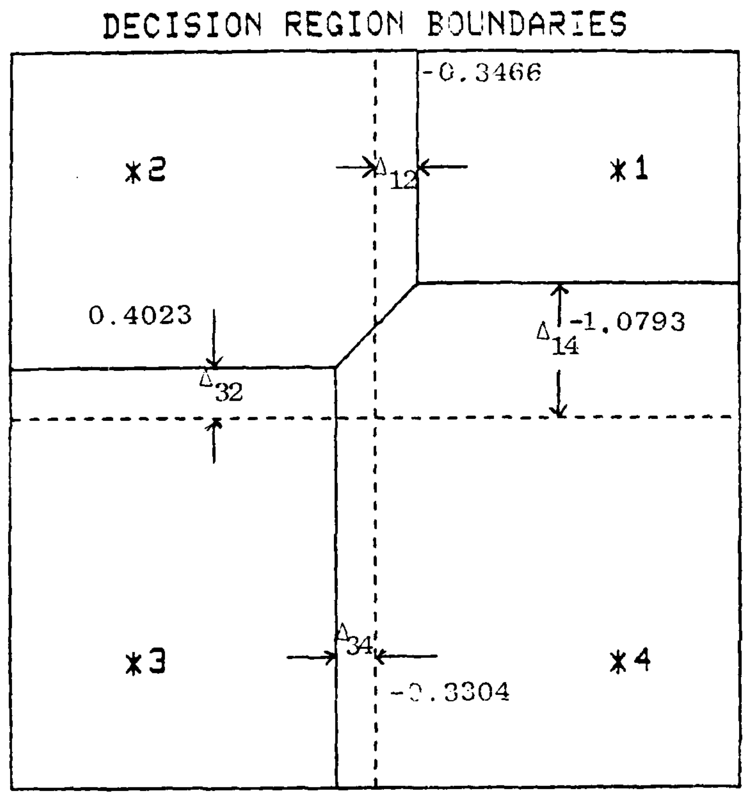


Fig 14. Decision Region Boundaries of Variation Two With Values of DELTA

By substitution, we find Δ_{12}

$$\Delta_{12} = \frac{2}{(2)(4)} \ln \frac{.01}{.04} = -0.3466$$

Similarly, we compute the remaining values:

$$\Delta_{14} = -1.0793$$
$$\Delta_{32} = 0.4023$$
$$\Delta_{34} = -0.3304$$

It has been shown that the boundary moves toward the signal of lesser probability and Figure 14 displays this fact and the amount four of the boundaries have been moved. In order to determine $P\{e|m_1\}$, we note that an error will be made if n_1 is greater than $(2 - 0.3466) = 1.6534$ or n_2 is greater than $(2 - 1.0793) = 0.9207$. Then as before

$$P\{e|m_1\} = Q[1.6534] + Q[0.9207] - Q[1.6534]*Q[0.9207]$$

$$P\{e|m_1\} = 0.2186$$

For signal three, we see an error will occur if n_1 is greater than $(2 - 0.3304) = 1.6696$ or if n_2 is greater than $(2 + 0.4023) = 2.4023$. Hence, we compute

$$P\{e|m_3\} = Q[1.6696] + Q[2.4023] - Q[1.6696]*Q[2.4023]$$

$$P\{e|m_3\} = 0.05532$$

Table VII provides the program returned results for this variation of the problem. By examination of the table, we first note that the signal probabilities have been appropriately changed, yet the energy and signal-to-noise ratios

TABLE VII

Statistics of Variation Two

OPTION [illegible]

SIGNAL	COORDINATES		PROBABILITY	ENERGY	SNR	UNION BOUND PROB OF ERROR
1	.20[illegible]E+01	.200E+01	.0100	.800E+01	6.0206	.2385E+00
2	-.2[illegible]E+01	.200E+01	.0400	.800E+01	[illegible].0206	.7497E-01
3	[illegible]E+01	-.200E+01	.2000	.800E+01	[illegible]	.5604E-01
4	.2[illegible]E+01	-.200E+01	.7500	.[illegible]0E+01	6.02[illegible]	.1134E-01

SYSTEM [illegible] PSD : 1.

TOTAL SY[illegible]M ENERGY : .320E+02

TOTAL SY[illegible]M PROBABILITY OF ERROR: < .25[illegible]E-01

SYSTEM [illegible]R OF GRAVITY LOCATED AT [illegible]304E+01 -.180E+01

OPTION [illegible]

SIGNAL	PROBABILITY	INTEGRATED PROB OF ERROR
1	.0100	.2190E+00
2	.0400	.6557E-01
3	.2000	.5526E-01
4	.7500	.1099E-01

TOTAL SY[illegible]M PROBABILITY OF ERROR: < .2411E-01

remain the same as the original problem. Comparison to the computations of the union bound on P_e shows them to be as close as round-off error will allow. Finally, comparison of the integrated P_e for signals one and three (found in OPTION> 12 of Table VII) with the actual P_e computed above, reveals the two differ by less than two-tenths of one percent.

Variation Three: Deletion of Signal. We return to the original problem and using OPTION> 3, signal s_4 is deleted from the signal set. Since the union bound is the only value not previously computed, we apply Eq (66) once again to determine the values below:

$$P\{e|m_1\} = 0.02515$$
$$P\{e|m_2\} = 0.0455$$
$$P\{e|m_3\} = 0.02515$$

Comparison with the values found in Table VIII shows the two computations to be nearly identical. The calculation of the actual P_e for signal s_2 has already been validated. For signals s_1 and s_3, however, the previous method of manual calculation is not possible. The necessity of integrating the inseparable joint density of the noise components prevents a comparison from being made. Note that in this case the integrated P_e is less than the union bound value for P_e; hence, the approximation obtained by integration provides a tighter bound.

TABLE VIII

Statistics of Variation Three

OPTION [illegible]

SIGNAL	COORDINATES		PROBABILITY	ENERGY	SNR	UNION BOUND PROB OF ERROR
1	-.200E+01	.200E+01	.3333	.800E+01	6.0206	.2509E-01
2	-.200E+01	-.200E+01	.3333	.800E+01	6.0206	.4550E-01
3	.200E+01	-.200E+01	.3333	.800E+01	6.0206	.2509E-01

SYSTEM NOISE PSD : 1.

TOTAL SYSTEM ENERGY : .240E+02

TOTAL SYSTEM PROBABILITY OF ERROR: < .3189E-01

SYSTEM CENTER OF GRAVITY LOCATED AT -.667E+00 -.667E+00

OPTION [illegible]

SIGNAL	PROBABILITY	INTEGRATED PROB OF ERROR
1	.3333	.2368E-01
2	.3333	.4498E-01
3	.3333	.2370E-01

TOTAL SYSTEM PROBABILITY OF ERROR: < .3079E-01

Variation Four: Addition of a Signal. Using OPTION> 5 allows a signal at (0,4) to be added.

1. Signal Energy. The energy of the newly added signal is computed from Eq (79) as

$$\text{Energy} = 0 + (4)^2 = 16 \text{ joules}$$

2. Signal-to-Noise Ratio. Similarly, we use Eq (78) to find the SNR for signal five.

$$\frac{E_{s_5}}{N_o} \Big| \text{db} = 10 \log_{10} \frac{16}{2} = 9.0309 \text{ db}$$

3. Union Bound on the Probability of Error. Since the signals are once again equiprobable, Eq (66) is used and provides the values below:

$$P\{e|m_1\} = 0.1269$$
$$P\{e|m_2\} = 0.0485$$
$$P\{e|m_3\} = 0.0485$$
$$P\{e|m_4\} = 0.1269$$
$$P\{e|m_5\} = 0.1592$$

Using Table IX for comparison of these statistics, once again the program returned values are validated.

4. For the reasons outlined in the previous variation, the calculation of the exact P_e is not done for this example.

TABLE IX

Statistics of Variation Four

OPTION

SIGNAL	COORDINATES		PROBABILITY	ENERGY	SNR	UNION BOUND PROB OF ERROR
1	-.[illegible]E+01	.200E+01	.2000	.800E+01	6.0206	.1265E+00
2	-.[illegible]E+01	-.200E+01	.2000	.800E+01	6.02[illegible]	.48[illegible]E-01
3	[illegible]E+01	-.200E+01	.2000	.800E+01	6.0206	.48[illegible]-01
4	.[illegible]E+01	.200E+01	.2000	.800E+01	6.02[illegible]	.1265E+00
5	0.	.400E+01	.2000	.160E+02	9.030[illegible]	.1589E+00

SYSTEM NOISE PSD : 1.

TOTAL SIGNAL ENERGY : .480E+02

TOTAL SYSTEM PROBABILITY OF ERROR: < .1[illegible]E+00

SYSTEM CENTER OF GRAVITY LOCATED AT 0. .800E+00

OPTION [illegible]

SIGNAL	PROBABILITY	INTEGRATED PROB OF ERROR
1	.2000	.1097E+00
2	.2000	.4498E-01
3	.2000	.4498E-01
4	.2000	.1097E+00
5	.2000	.1523E+00

TOTAL SYSTEM PROBABILITY OF ERROR: < .9232E-01

Analysis of Integrated Probability of Error

In order to demonstrate the effectiveness of the program in providing a reliable approximation of the actual probability of error, two conditions must be met. First, the value of P_e returned for any given signal must be insensitive to rotation of the signal set. Translation of the signal set is not relevant since prior to any computation, the signal is translated to the origin as described in Chapter III. Secondly, for reasons already discussed, we require the calculated value to be greater than or equal to the actual probability of error. The method chosen to demonstrate the successful achievement of these criterion simultaneously is as follows:

* The integrated P_e for each signal in the signal set of Figure 12 is determined
* The signal set is rotated 15 degrees
* The integrated P_e is again computed
* This procedure is continued a total of six times resulting in a total rotation of 90 degrees

The test results are presented in Table X. The first row gives the statistics for the original signal set as presented earlier in Table V. The subsequent rows provide the results for successive 15° rotations. As can be seen, the computed probability does vary as the signal set is rotated, but the largest deviation from the actual value is still less than 1.6 percent. More importantly is the fact that the values

TABLE X

Verification of Integrated Probability of Error

Angle (deg)	$Pe(s_1)$	$Pe(s_2)$	$Pe(s_3)$	$Pe(s_4)$
0	0.04498	0.04498	0.04498	0.04498
15	0.04569	0.04552	0.04565	0.04556
30	0.04561	0.04549	0.04560	0.04554
45	0.04558	0.04549	0.04558	0.04552
60	0.04560	0.04549	0.04561	0.04554
75	0.04565	0.04552	0.04569	0.04559
90	0.04498	0.04498	0.04536	0.04498
Maximum Deviation	0.00071	0.00054	0.00071	0.00061
Percent Deviation	1.578	1.200	1.578	1.356

returned always slightly over bound the actual probability of error as desired, but never more than 1.6 percent.

Manipulation of Graphic Display

This part of the verification process demonstrates that the graphical presentation actually displays the signal set and its decision region boundaries. Furthermore, we want to be able to translate, rotate, and scale the signal set and examine the results. It was explained in Chapter III that these operations consist of appropriately computing new signal point coordinates and performing all required calculations on the altered signal set. The mathematical procedure does not change, hence, it is not discussed here; however, the graphical representation of the signal set does change. The best approach to verify this ability, then, is to provide actual program generated plots. We begin with the signal set shown in Figure 15. Table XI lists the signal coordinates and the system statistics. Figure 16 displays the signal set with the decision region boundaries added.

Translation. Using OPTION> 5, the signal set is translated five units in both directions. Figure 17 displays the results of this translation and Table XII provides the statistics of the altered signal set. We note that the signal coordinates, hence the signal energies and signal-to-noise ratios, have been adjusted according to their new

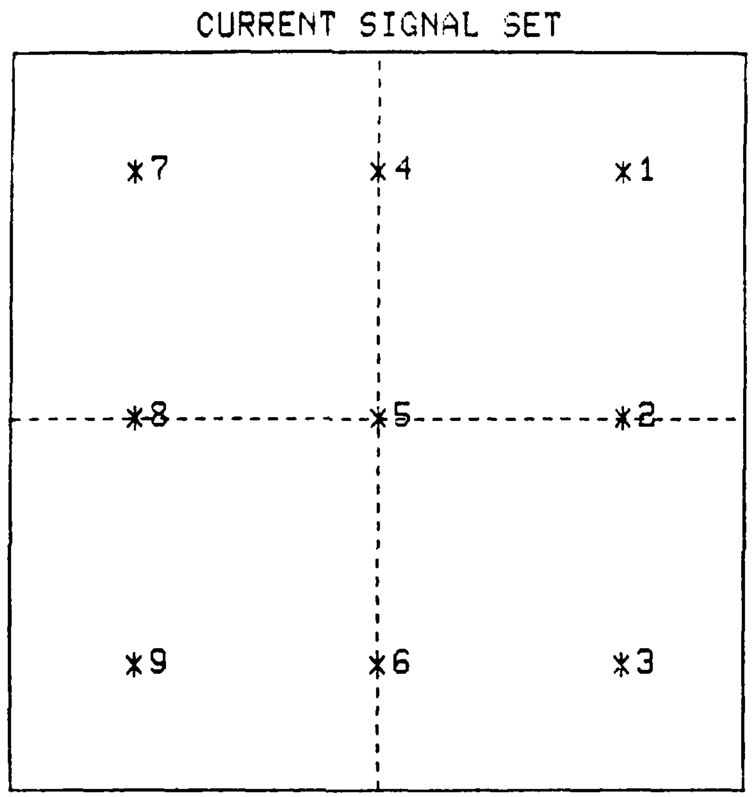


Fig 15. Nine Point Signal Set for Graphical Verification

TABLE XI

Statistics of Nine Point Signal Set

OPTION [illegible]

SIGNAL	COORDINATES		PROBABILITY	ENERGY	SNR	UNION BOUND PROB OF ERROR
1	[illegible]+01	.200E+01	.1111	[illegible]+01	6.0206	.4691E+00
2	[illegible]E+01	0.	.1111	.400E+01	3.0103	.6814E+00
3	[illegible]E+01	-.200E+01	.1111	.800E+01	6.0206	.4691E+00
4	0.	.200E+01	.1111	.400E+01	3.01[illegible]	.681[illegible]E+00
5	0.	0.	.1111	0.	-999.9999	.9492E+00
6	0.	-.200E+01	.1111	.400E+01	3.0103	.6814E+00
7	-.[illegible]+01	.200E+01	.1111	.800E+01	6.0206	.4691E+00
8	-.[illegible]E+01	0.	.1111	.400E+01	3.0103	.6814E+00
9	-.[illegible]E+01	-.200E+01	.1111	.800E+01	6.0206	.4691E+00

SYSTEM NOISE PSD : 1.

TOTAL SYSTEM ENERGY : .480E+02

TOTAL SYSTEM PROBABILITY OF ERROR: < .6168E+00

SYSTEM CENTER OF GRAVITY LOCATED AT 0. 0.

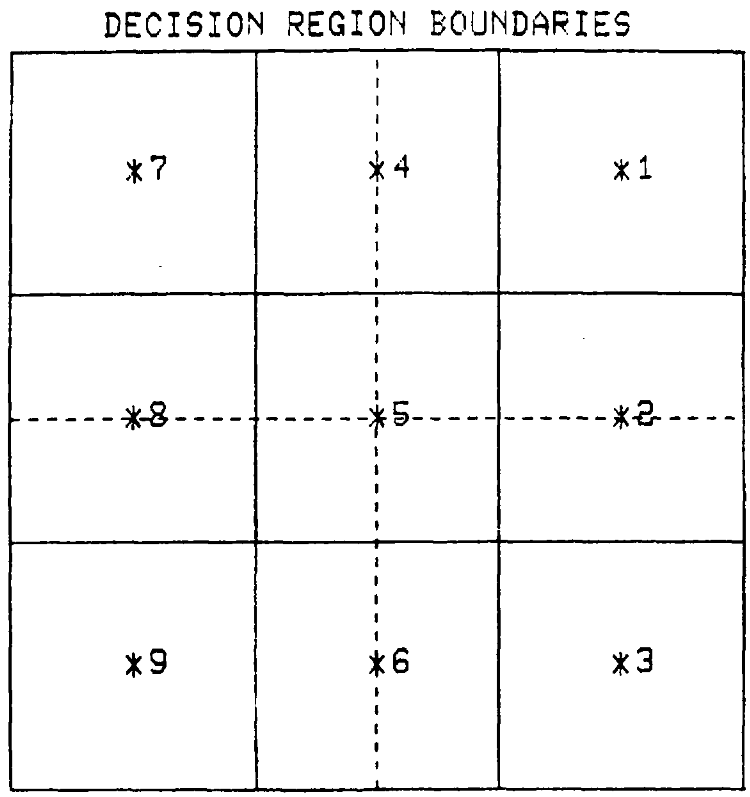


Fig 16. Decision Region Boundaries of Nine Point Signal Set

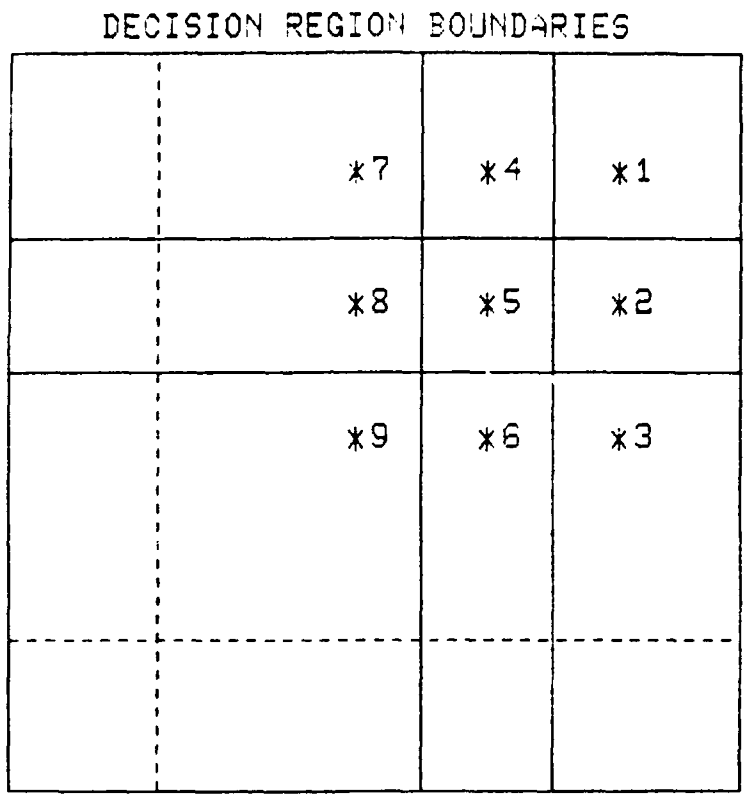


Fig 17. Translated Nine Point Signal Set

TABLE XII

Statistics of Translated Nine Point Signal Set

OPTION [illegible]

SIGNAL	COORDINATES		PROBABILITY	ENERGY	SNR	UNION BOUND PROB OF ERROR
1	.700E+01	.700E+01	.1111	.980E+02	16.9029	.4591E+00
2	.700E+01	.500E+01	.1111	.740E+02	15.6820	.6814E+00
3	.700E+01	.300E+01	.1111	.580E+02	14.6240	.4691E+00
4	.500E+01	.700E+01	.1111	.740E+02	15.6820	.6814E+00
5	.500E+01	.500E+01	.1111	.500E+02	13.9794	.9492E+00
6	.500E+01	.300E+01	.1111	.340E+02	12.3045	.6814E+00
7	.300E+01	.700E+01	.1111	.580E+02	14.6240	.4691E+00
8	.300E+01	.500E+01	.1111	.340E+02	12.3045	.6814E+00
9	.300E+01	.300E+01	.1111	.180E+02	9.5424	.4691E+00

SYSTEM NOISE PSD : 1.

TOTAL SYSTEM ENERGY : .498E+03

TOTAL SYSTEM PROBABILITY OF ERROR: < .6169E+00

SYSTEM CENTER OF GRAVITY LOCATED AT .500E+01 .500E+01

values. The computation of the union bound on P_e is unchanged because the parameters used in its calculation have not been changed by the translation.

Rotation. The signal set is translated to its original location and OPTION> 2 is used to rotate it 45 degrees. The resulting display is provided in Figure 18 and its statistics are located in Table XIII. The outcome is as expected. Note, these operations can be performed in any order in succession and as an example, the rotated signal set is translated. Figure 19 and Table XIV are the result.

Scaling. As described in Chapter III, there may be occasions when the user may want the window boundaries moved out, in effect expanding the field of view. In order to demonstrate this capability, the original nine point signal set is used and OPTION> 11 called to enter a scaling factor of two. Figure 20 provides the outcome of this operation.

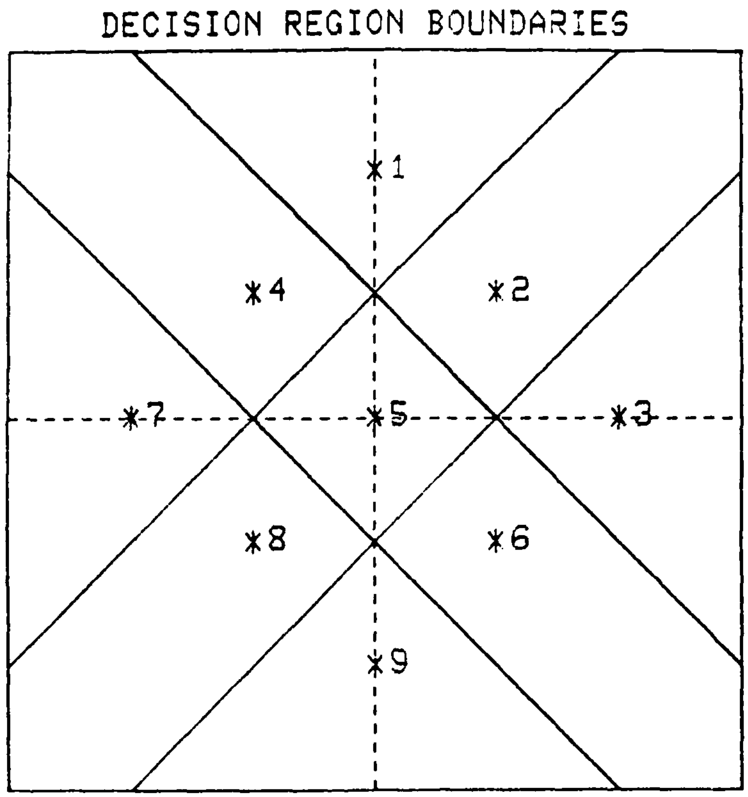


Fig 18. Rotated Nine Point Signal Set

TABLE XIII

Statistics of Rotated Nine Point Signal Set

OPTION [illegible]

SIGNAL	COORDINATES		PROBABILITY	ENERGY	SNR	UNION BOUND PROB OF ERROR
1	[illegible]-07	.283E+01	.1111	[illegible]	6.020[illegible]	[illegible]E+00
2	[illegible]+01	.141E+01	.1111	.400[illegible]+01	3.01[illegible]	.681[illegible]E+00
3	[illegible]+01	-.379E-07	.1111	.800[illegible]	6.0[illegible]	[illegible]
4	[illegible]+01	.141E+01	.1111	[illegible]	3.010[illegible]	[illegible]
5	0.	0.	.1111	0.	-999.99[illegible]	.99[illegible]E+00
6	[illegible]+01	-.141E+01	.1111	.400[illegible]+01	3.010[illegible]	.681[illegible]E+00
7	[illegible]+01	.379E-07	.1111	.800[illegible]+01	6.020[illegible]	[illegible]
8	[illegible]+01	-.141E+01	.1111	.400[illegible]	3.0103	.681[illegible]E+00
9	[illegible]-07	-.283E+01	.1111	.800E+01	6.020[illegible]	.4691E+00

SYSTEM [illegible] PSD : 1.

TOTAL SYSTEM ENERGY : .400E+02

TOTAL SYSTEM PROBABILITY OF ERROR: < .6162E+00

SYSTEM CENTER OF GRAVITY LOCATED AT -.987E-16 0.

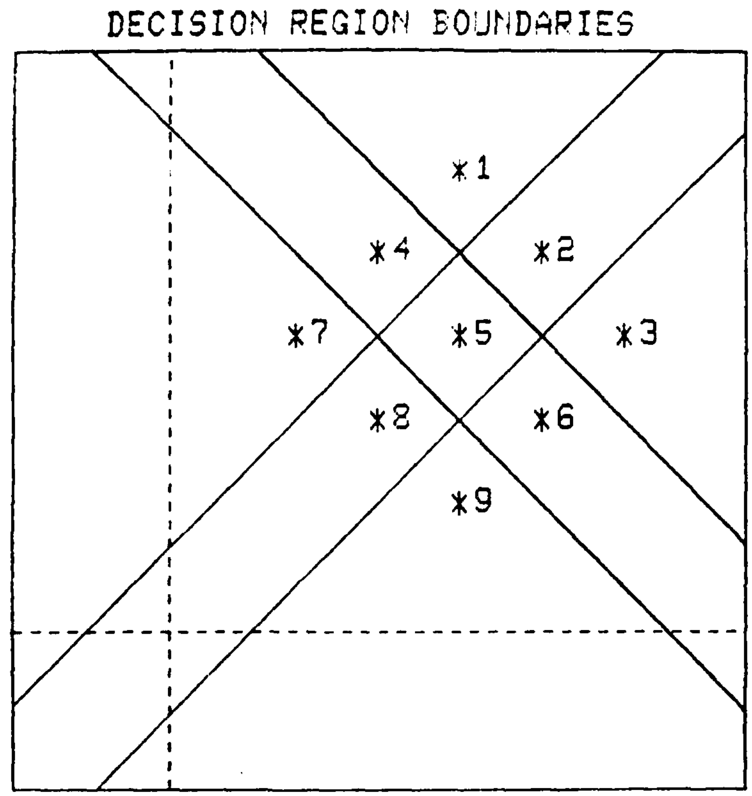


Fig 19. Rotated and Translated Nine Point Signal Set

TABLE XIV

Statistics of Rotated and Translated Nine Point Signal Set

OPTION [illegible]

SIGNAL	COORDINATES		PROBABILITY	ENERGY	SNR	UNION BOUND PROB OF ERROR
1	[illegible]+01	.283E+01	.1111	.863E+02	16.3490	.4691E+00
2	[illegible]+01	.641E+01	.1111	.823E+02	16.14[illegible]	.60[illegible]E+00
3	[illegible]+01	.500E+01	.1111	.863E+02	16.3490	.46[illegible]
4	[illegible]+01	.641E+01	.1111	.540E+02	14.47[illegible]	.681[illegible]
5	[illegible]+01	.500E+01	.1111	.500E+02	13.97[illegible]	.949[illegible]
6	[illegible]+01	.359E+01	.1111	.540E+02	14.31[illegible]	.581[illegible]
7	[illegible]+01	.500E+01	.1111	.29[illegible]	[illegible]	.959[illegible]E+00
8	[illegible]+01	.359E+01	.1111	.297E+02	11.09[illegible]	[illegible]
9	[illegible]+01	.217E+01	.1111	.297E+02	11.71[illegible]	.48[illegible]

SYSTEM NOISE PSD : 1.

TOTAL SYSTEM ENERGY : .498E+03

TOTAL SYSTEM PROBABILITY OF ERROR: < .6168E+00

SYSTEM CENTER OF GRAVITY LOCATED AT .500E+01 .500E+01

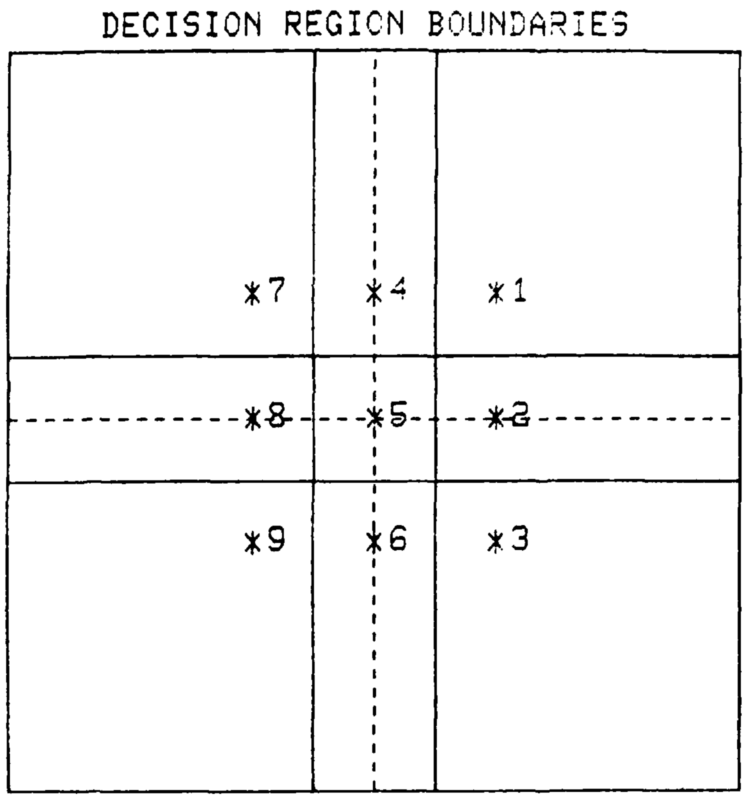


Fig 20. Display of Nine Point Signal Set Scaled By Two

V. Conclusions and Recommendations

In this project, a computer program was developed to interactively solve problems relating to signal detection theory. Use of the Tektronix PLOT-10 graphics package provides visual display of the problems being solved. In particular, the user selects the signal set and specifies the channel parameters, then the program displays the resulting decision region boundaries and computes the statistics of each signal and the system in general. The results of varying any parameter can be quickly determined and displayed. As an aid in studying signal detection theory, the program allows the user to investigate the interactions of the system parameters without having to perform hours of tedious computations. The user can thus learn how proper selection of signal set and system parameters can optimize the probability of correctly receiving transmitted messages.

The program may be executed from any interactive computer terminal which is supported by the FORTRAN 77 compiler and has access to the International Mathematical and Statistical Library (IMSL) routines MDNOR and MDBNOR. However, for visual displays, the use of Tektronix terminals model 4014, 4012, or 4010 is essential. The ability to plot the decision region boundaries at any interactive

terminal would increase the flexibility and use of the program. It is recommended that the feasibility of adding this plotting capability be pursued.

Secondly, the program as designed maintains the user supplied input data in temporary core storage only. Thus, each time the program is executed, a signal set must be made. This is not a severe limitation when working with small signal sets, i.e., those with less than ten signal points. However, when working with larger signal sets, the amount of time required to specify the signal set linearly increases. The capability to file these larger signal sets outside the main program and to be able to attach the file for use later, would be beneficial to those doing analysis of a particular signal set over an extended period of time. It is suggested, therefore, that the feasibility of such an alteration be studied and this capability added to the existing program.

As a final note, the current configuration of the program fully utilizes the allocated dynamic storage capacity of the interactive system (CYBER INTERCOM). Thus, in order to implement either of the proposals above, the program would have to be restructured to use overlays or the amount of allocated dynamic storage space would have to be increased.

Bibliography

1. Davis, Richard M. Thesis Projects in Science and Engineering. New York: St. Martin's Press, 1980.

2. Helstrom, Carl W. Statistical Theory of Signal Detection. New York: Pergamon Press, 1960.

3. Melsa, James L. and David L. Cohn. Decision and Estimation Theory. New York: McGraw-Hill Book Company, 1978.

4. Middlemiss, Ross R., et al. Analytic Geometry. New York: McGraw-Hill Book Company, 1968.

5. Middleton, David. An Introduction to Statistical Communication Theory. New York: McGraw-Hill Book Company, 1960.

6. PLOT-10. Tektronix PLOT-10/Terminal Control System. User's Manual. Beaverton, Oregon: Tektronix, Inc., 1974.

7. VanTrees, Harry L. Detection, Estimation, and Modulation Theory, Part I. New York: John Wiley and Sons, Inc., 1968.

8. Wagener, Jerrold L. FORTRAN 77 Principles of Programming. New York: John Wiley and Sons, Inc., 1980.

9. Weast, Robert C., et al. Handbook of Tables for Mathematics (Fourth Edition). Cleveland, Ohio: Chemical Rubber Company, 1970.

10. Wozencraft, J. M. and I. M. Jacobs. Principles of Communication Engineering. New York: John Wiley and Sons, Inc., 1965.

Appendix A: Supporting Derivations

Derivation I Derivation of the Mean and Variance of the Sufficient Statistic of the General Gaussian Problem

The sufficient statistic as determined in the text of Chapter II (Eq (24)) is given by

$$\ell(\underline{z}) = (\underline{s}_2 - \underline{s}_1)^T \underline{V}^{-1} \underline{z} \tag{24}$$

Since $\underline{z}_k = \underline{s}_k + \underline{n}$

The expectation of ℓ given message k becomes

$$E\{\ell|m_k\} = E[(\underline{s}_2 - \underline{s}_1)^T \underline{V}^{-1} (\underline{s}_k + \underline{n})] \tag{A-1}$$

Since $(\underline{s}_2 - \underline{s}_1)^T \underline{V}^{-1}$ is a constant and $E\{n\} = 0$, we have

$$\begin{aligned} E\{\ell|m_k\} &= E[(\underline{s}_2 - \underline{s}_1)^T \underline{V}^{-1} \underline{s}_k] \\ &= (\underline{s}_2 - \underline{s}_1)^T \underline{V}^{-1} \underline{s}_k \end{aligned} \tag{A-2}$$

The variance is computed from

$$\mathrm{Var}\{\ell|m_k\} = E\{\ell^2|m_k\} - [E\{\ell|m_k\}]^2 \tag{A-3}$$

where $E\{\ell|m_k\}$ is given by Eq (A-2) and $E\{\ell^2|m_k\}$ is determined below.

$$E\{\ell^2|m_k\} = E\{(\underline{s}_2 - \underline{s}_1)^T \underline{V}^{-1} (\underline{s}_k + \underline{n})$$
$$(\underline{s}_2 - \underline{s}_1)^T \underline{V}^{-1} (\underline{s}_k + \underline{n})\}$$

$$= E\{(\underline{s}_2 - \underline{s}_1)^T \underline{V}^{-1} (\underline{s}_k + \underline{n})$$
$$(\underline{s}_k + \underline{n})^T \underline{V}^{-1} (\underline{s}_2 - \underline{s}_1)\} \qquad \text{(A-4)}$$

Multiplying out the two center terms yields

$$(\underline{s}_k + \underline{n})\ (\underline{s}_k + \underline{n})^T = \underline{s}_k\ \underline{s}_k^{\ T} + \underline{s}_k\ \underline{n}^T + \underline{n}\ \underline{s}_k^{\ T} + \underline{n}\ \underline{n}^T$$

Hence, the expectation of this product can be written

$$E\{(\underline{s}_k + \underline{n})\ (\underline{s}_k + \underline{n})^T\} = E\{\underline{s}_k\ \underline{s}_k^{\ T} + \underline{s}_k\ \underline{n}^T + \underline{n}\ \underline{s}_k^{\ T} + \underline{n}\ \underline{n}^T\}$$

And since $E\{\underline{n}\} = 0$ and $E\{\underline{n}\ \underline{n}^T\} = \underline{V}$, the expectation of the product becomes

$$E\{(\underline{s}_k + \underline{n})\ (\underline{s}_k + \underline{n})^T\} = \underline{s}_k\ \underline{s}_k^{\ T} + \underline{V} \qquad \text{(A-5)}$$

By substitution into Eq (A-4)

$$E\{\ell^2|m_k\} = (\underline{s}_2 - \underline{s}_1)^T \underline{V}^{-1} (\underline{s}_k\ \underline{s}_k^{\ T} + \underline{V})$$
$$\underline{V}^{-1} (\underline{s}_2 - \underline{s}_1) \qquad \text{(A-6)}$$

Multiplying this out yields

$$E\{\ell^2|m_k\} = [(\underline{s}_2 - \underline{s}_1)^T \underline{V}^{-1}\ \underline{s}_k\ \underline{s}_k^{\ T} + (\underline{s}_2 - \underline{s}_1)^T \underline{V}^{-1}\ \underline{V}]$$
$$\underline{V}^{-1} (\underline{s}_2 - \underline{s}_1)$$

$$= (\underline{s}_2 - \underline{s}_1)^T \underline{V}^{-1} \underline{s}_k \underline{s}_k^T \underline{V}^{-1} (\underline{s}_2 - \underline{s}_1)$$

$$+ (\underline{s}_2 - \underline{s}_1)^T \underline{V}^{-1} (\underline{s}_2 - \underline{s}_1)$$

$$= [(\underline{s}_2 - \underline{s}_1)^T \underline{V}^{-1} \underline{s}_k] [(\underline{s}_2 - \underline{s}_1)^T \underline{V}^{-1} \underline{s}_k]$$

$$+ (\underline{s}_2 - \underline{s}_1)^T \underline{V}^{-1} (\underline{s}_2 - \underline{s}_1)$$

$$= [(\underline{s}_2 - \underline{s}_1)^T \underline{V}^{-1} \underline{s}_k]^2$$

$$+ (\underline{s}_2 - \underline{s}_1) \underline{V}^{-1} (\underline{s}_2 - \underline{s}_1) \quad \text{(A-7)}$$

Substitution of Eqs (A-2) and (A-7) into Eq (A-3) results in

$$\mathrm{Var}\{\ell | m_k\} = (\underline{s}_2 - \underline{s}_1)^T \underline{V}^{-1} (\underline{s}_2 - \underline{s}_1) \quad \text{(A-8)}$$

Using the definition for $\Delta\underline{s}$ as in Chapter II

$$\Delta\underline{s} = \underline{s}_2 - \underline{s}_1$$

We conclude by substitution into Eq (A-2) to obtain the simplified means

$$E\{\ell | m_1\} = \Delta\underline{s}^T \underline{V}^{-1} \underline{s}_1$$

$$E\{\ell | m_2\} \quad \Delta\underline{s}^T \underline{V}^{-1} \underline{s}_2 \quad \text{(A-9)}$$

And the simplified variance

$$\mathrm{Var}\{\ell | m_k\} = \Delta\underline{s}^T \underline{V}^{-1} \Delta\underline{s} \quad \text{for} \quad k = 1, 2, \ldots, I \quad \text{(A-10)}$$

Derivation II Derivation of the Probability of False Alarm Eq (30)

We begin with the probability of false alarm as determined in Chapter II by Eq (29) which is provided again below.

$$P_{FA} = \int_{\lambda'}^{\infty} \frac{1}{(2\pi)^{\frac{1}{2}} \, (\Delta\underline{s}^T \, \underline{V}^{-1} \, \Delta\underline{s})^{\frac{1}{2}}} \exp\left[- \frac{1}{2} \frac{(\ell - \Delta\underline{s}^T \, \underline{V}^{-1} \, \underline{s}_1)^2}{\Delta\underline{s}^T \, \underline{V}^{-1} \, \Delta\underline{s}} \right] d\ell \qquad (29)$$

If we let $\beta = \dfrac{\ell - \Delta\underline{s}^T \, \underline{V}^{-1} \, \underline{s}_1}{(\Delta\underline{s}^T \, \underline{V}^{-1} \, \Delta\underline{s})^{\frac{1}{2}}}$

Then its derivative becomes

$$d\beta = \frac{d\ell}{(\Delta\underline{s}^T \, \underline{V}^{-1} \, \Delta\underline{s})^{\frac{1}{2}}}$$

Now by substitution and appropriate adjustment of the limits of integration, P_{FA} becomes

$$P_{FA} = \int_{\frac{\lambda' - \Delta\underline{s}^T \, \underline{V}^{-1} \, \underline{s}_1}{(\Delta\underline{s}^T \, \underline{V}^{-1} \, \Delta\underline{s})^{\frac{1}{2}}}}^{\infty} \frac{1}{\sqrt{2\pi}} \exp\left[- \frac{\beta^2}{2} \right] d\beta \qquad \text{(A-11)}$$

Using the Q-function transformation

$$P_{FA} = Q\left[\frac{\lambda' - \Delta\underline{s}^T\ \underline{V}^{-1}\ \underline{s}_1}{(\Delta\underline{s}^T\ \underline{V}^{-1}\ \Delta\underline{s})^{\frac{1}{2}}}\right] \qquad (A-12)$$

From Eq (24) of Chapter II, we know

$$\lambda' = \ln\lambda + \frac{1}{2}\ [\underline{s}_2^T\ \underline{V}^{-1}\ \underline{s}_2 - \underline{s}_1^T\ \underline{V}^{-1}\ \underline{s}_1] \qquad (24)$$

Substitution into Eq (A-12) gives

$$P_{FA} = Q\left[\frac{\ln\lambda + \frac{1}{2}\underline{s}_2^T\ \underline{V}^{-1}\ \underline{s}_2 - \frac{1}{2}\underline{s}_1\ \underline{V}^{-1}\underline{s}_1 - (\underline{s}_2 - \underline{s}_1)^T\ \underline{V}^{-1}\ \underline{s}_1}{(\Delta\underline{s}^T\ \underline{V}^{-1}\ \Delta\underline{s})^{\frac{1}{2}}}\right] \qquad (A-13)$$

And because

$$(\underline{s}_2 - \underline{s}_1)^T\ \underline{V}^{-1}\ \underline{s}_1 = \underline{s}_2^T\ \underline{V}^{-1}\ \underline{s}_1 - \underline{s}_1^T\ \underline{V}^{-1}\ \underline{s}_1$$

Substituting again gives

$$P_{FA} =$$

$$Q\left[\frac{\ln\lambda + \frac{1}{2}\underline{s}_2^T\ \underline{V}^{-1}\ \underline{s}_2 + \frac{1}{2}\underline{s}_1\ \underline{V}^{-1}\ \underline{s}_1 - \frac{1}{2}\underline{s}_2^T\ \underline{V}^{-1}\ \underline{s}_1 - \frac{1}{2}\underline{s}_1^T\ \underline{V}^{-1}\ \underline{s}_2}{(\Delta\underline{s}^T\ \underline{V}^{-1}\ \Delta\underline{s})^{\frac{1}{2}}}\right] \qquad (A-14)$$

Using matrix algebra and combining like-terms yields

$$P_{FA} = Q\left[\frac{\ln\lambda + \frac{1}{2}\ (\underline{s}_2 - \underline{s}_1)^T\ \underline{V}^{-1}\ (\underline{s}_2 - \underline{s}_1)}{(\Delta\underline{s}^T\ \underline{V}^{-1}\ \Delta\underline{s})^{\frac{1}{2}}}\right] \qquad (A-15)$$

The definition for $\Delta\underline{s}$ as

$$\Delta\underline{s} = (\underline{s}_2 - \underline{s}_1)$$

Provides Eq (30) as desired.

$$P_{FA} = Q\left[\frac{\ln\lambda}{(\Delta\underline{s}^T\ \underline{V}^{-1}\ \Delta\underline{s})^{\frac{1}{2}}} + \frac{(\Delta\underline{s}^T\ \underline{V}^{-1}\ \Delta\underline{s})^{\frac{1}{2}}}{2}\right] \quad (30)$$

Derivation III Derivation of the Simplified Expression for the Union Bound, Eq (66)

We begin with Eq (63), rewritten below

$$\Pr\{\|\ \underline{z} - \underline{s}_k\ \|^2 > \|\ \underline{z} - \underline{s}_\ell\ \|^2 | m_k\}$$

$$= \Pr\{2\underline{n}^T\ (\underline{s}_\ell - \underline{s}_k) > \|\ \underline{s}_\ell - \underline{s}_k\ \|^2\} \quad (63)$$

Let

$$\beta = 2\underline{n}^T\ (\underline{s}_\ell - \underline{s}_k) = 2(\underline{s}_\ell - \underline{s}_k)^T\ \underline{n}$$

$$= \sum_{i=1}^{I} 2n_i\ [s_{\ell_i} - s_{k_i}] \quad (A\text{-}16)$$

We take the expectation of β to determine its mean, since,

$$E\{n_i\} = 0 \quad , \quad E\{\beta\} = 0 \quad (A\text{-}17)$$

Because the mean of β is zero, its variance is computed from

$$\mathrm{Var}\{\beta\} = E\{\beta^2\}$$

Which becomes

$$= E\{[2\underline{n}^T (\underline{s}_\ell - \underline{s}_k)]^T [2\underline{n}^T (\underline{s}_\ell - \underline{s}_k)]\}$$

$$= E\{[4 (\underline{s}_\ell - \underline{s}_k)^T \underline{n}\ \underline{n}^T (\underline{s}_\ell - \underline{s}_k)]\}$$

$$= 4 (\underline{s}_\ell - \underline{s}_k) E\{\underline{n}\ \underline{n}^T\} (\underline{s}_\ell - \underline{s}_k) \tag{A-18}$$

Now, since

$$E\{\underline{n}\ \underline{n}^T\} = \sigma^2 I \tag{A-19}$$

We get

$$\mathrm{Var}\{\beta\} = 4\sigma^2 (\underline{s}_\ell - \underline{s}_k)^T (\underline{s}_\ell - \underline{s}_k)$$

$$= 4\sigma^2 \| \underline{s}_\ell - \underline{s}_k \|^2 \tag{A-20}$$

By substitution of Eq (A-16) into Eq (63), we find

$$\Pr\{e|m_k\} = \Pr\{\| \underline{z} - \underline{s}_k \|^2 > \| \underline{z} - \underline{s}_\ell \|^2 | m_k\}$$

$$= \Pr\{\beta > \| \underline{s}_\ell - \underline{s}_k \|^2\} \tag{A-21}$$

Since β has been shown to be gaussian with zero mean and variance given by Eq (A-20), the conditional probability of error can be written

$$\Pr\{e|m_k\} = \int_{\|\underline{s}_\ell - \underline{s}_k\|^2}^{\infty} \frac{1}{\sqrt{2\pi \; 4\sigma^2 \|\underline{s}_\ell - \underline{s}_k\|^2}}$$

$$\exp\left[-\frac{\beta^2}{2[4\sigma^2\|\underline{s}_\ell - \underline{s}_k\|^2]}\right] d\beta \qquad \text{(A-22)}$$

Let

$$\gamma = \frac{\beta}{2\sigma\|\underline{s}_\ell - \underline{s}_k\|}$$

Then similar to the procedure of Derivation II, substitution gives

$$\Pr\{e|m_k\} = \int_{\frac{\|\underline{s}_\ell - \underline{s}_k\|}{2\sigma}}^{\infty} \frac{1}{\sqrt{2\pi}} \, e^{-\frac{\gamma^2}{2}} \, d\gamma \qquad \text{(A-23)}$$

Which can be transformed to the Q-function form to become

$$\Pr\{e|m_k\} = Q\left[\frac{\|\underline{s}_\ell - \underline{s}_k\|}{2\sigma}\right] \qquad \text{(A-24)}$$

Allowing $d_{\ell k} = \|\underline{s}_\ell - \underline{s}_k\|$, as before, yields the form of Eq (66)

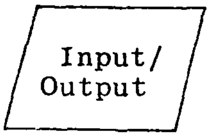
$$P\{e|m_k\} = Q\left[\frac{d_{\ell k}}{2\sigma}\right] \qquad \text{(A-25)}$$

APPENDIX B

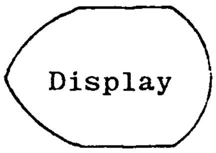
Flow Charts

This appendix presents the flow charts for the main program and the 23 subroutines. To prevent confusion, the symbols used are defined below.

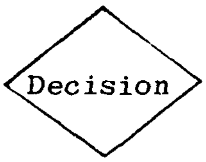
Any processing function causing a change in value



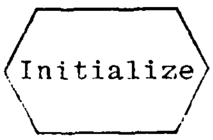
General input/output function



Information displayed via graphics subroutines

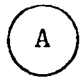


Decision operation



Initialization of parameters or variables

Beginning or ending of subroutine. The routine name is used at the start and "RETURN" is used at the end.



On page connector, letters used.

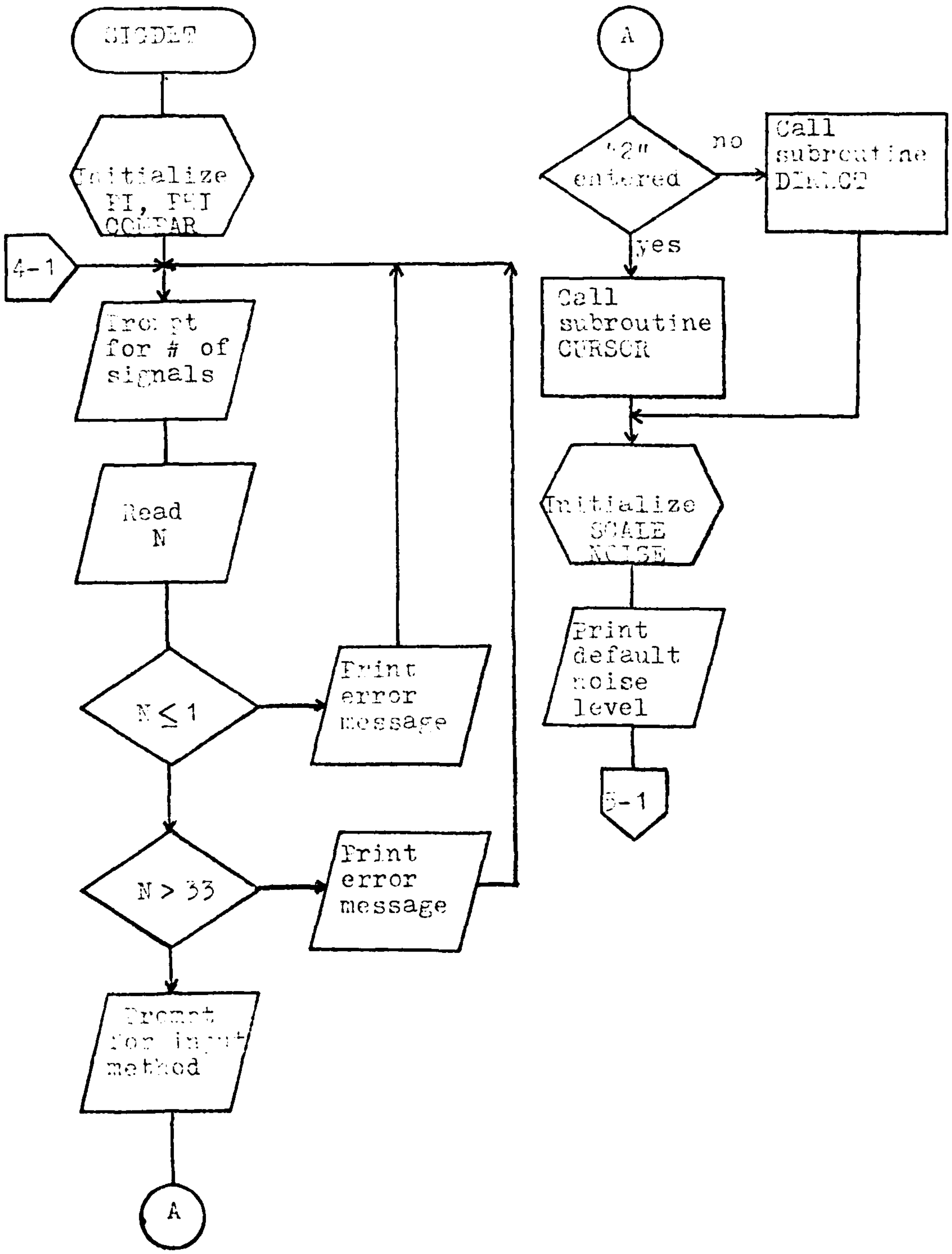
Off page connector. The first number indicates page of Appendix B where routine is continued, and the second indicates location on the page.

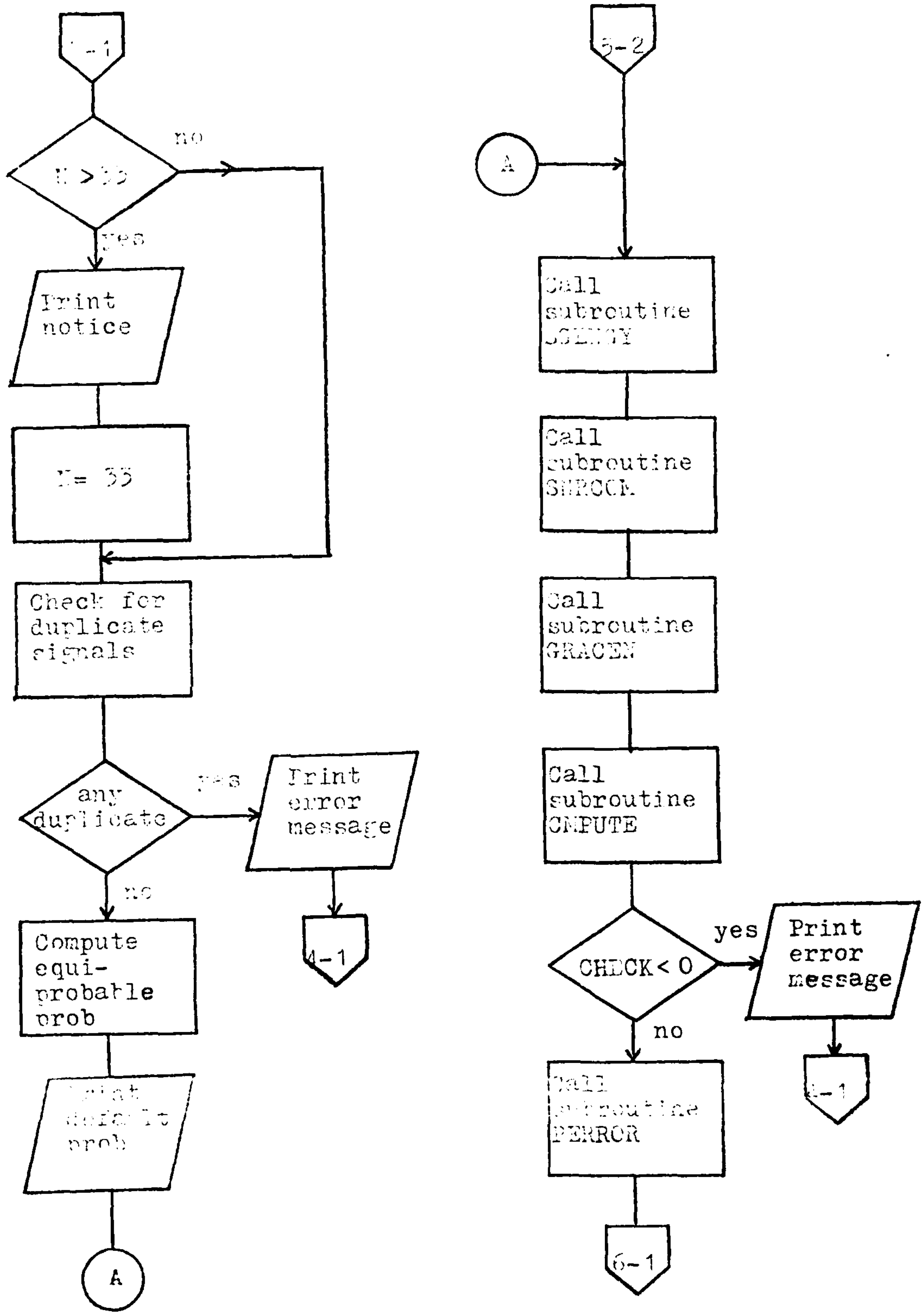
In addition to normal abbreviations, the following special abbreviations are used:

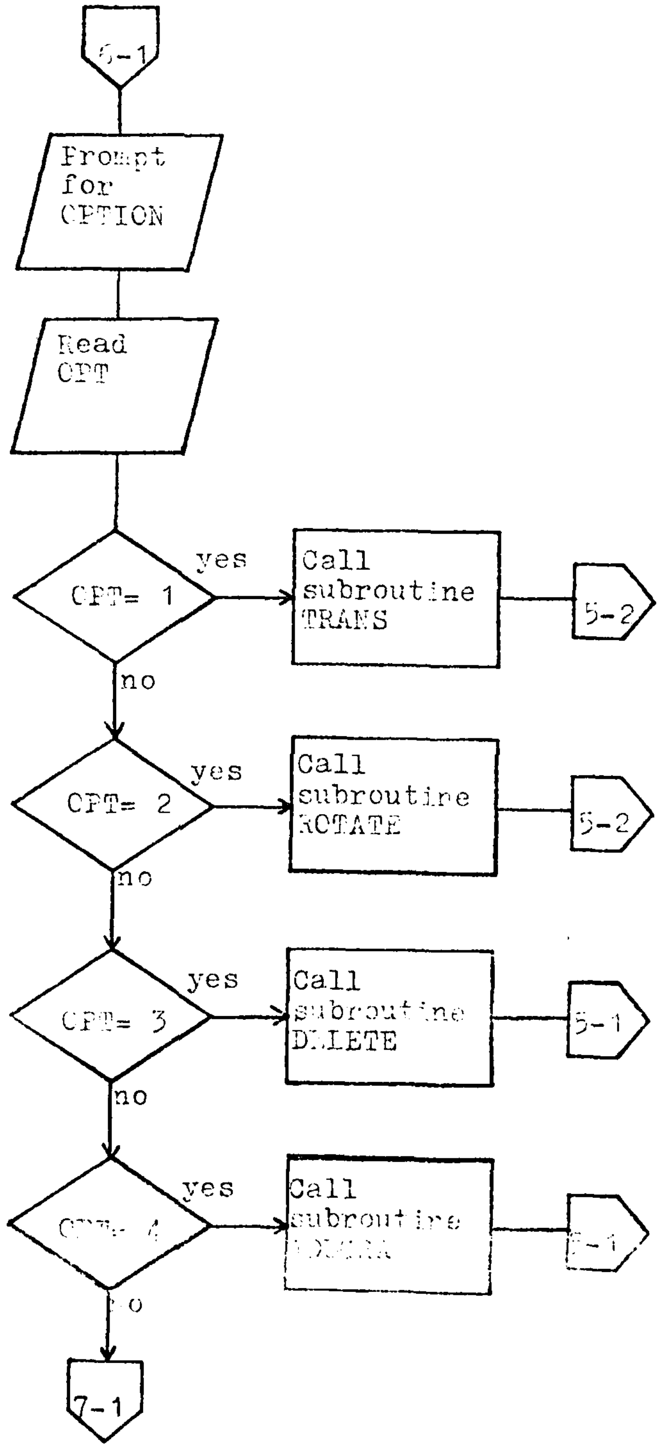
CNT	Connecting or Connector
COORD	Coordinate(s)
HORT	Horizontal
INSTRCT	Instruction(s)
INTRSCT	Intersection
LD PT	Lead Point
POI	Point of Intersection
PROB	Probability
VERT	Vertical

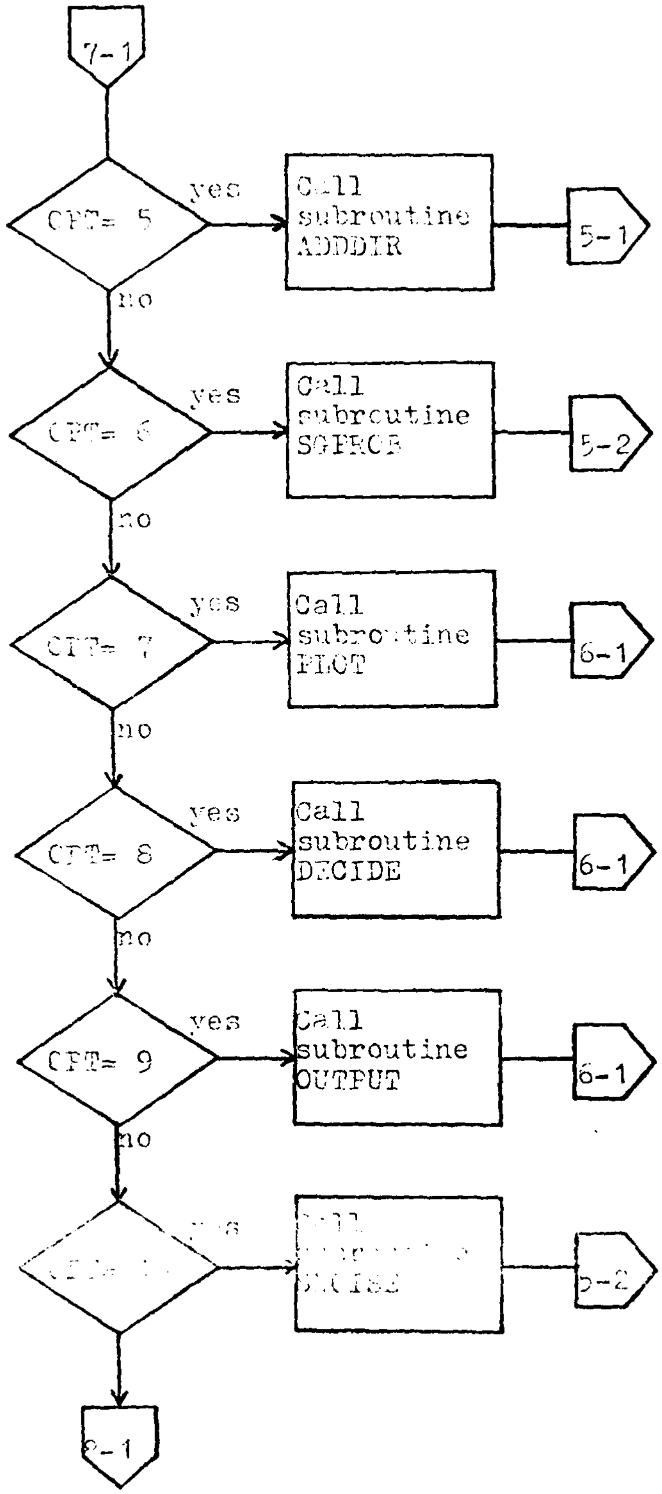
The index located on the next page is included to facilitate locating the flowcharts for the individual routines.

1. Main Program SIGDET

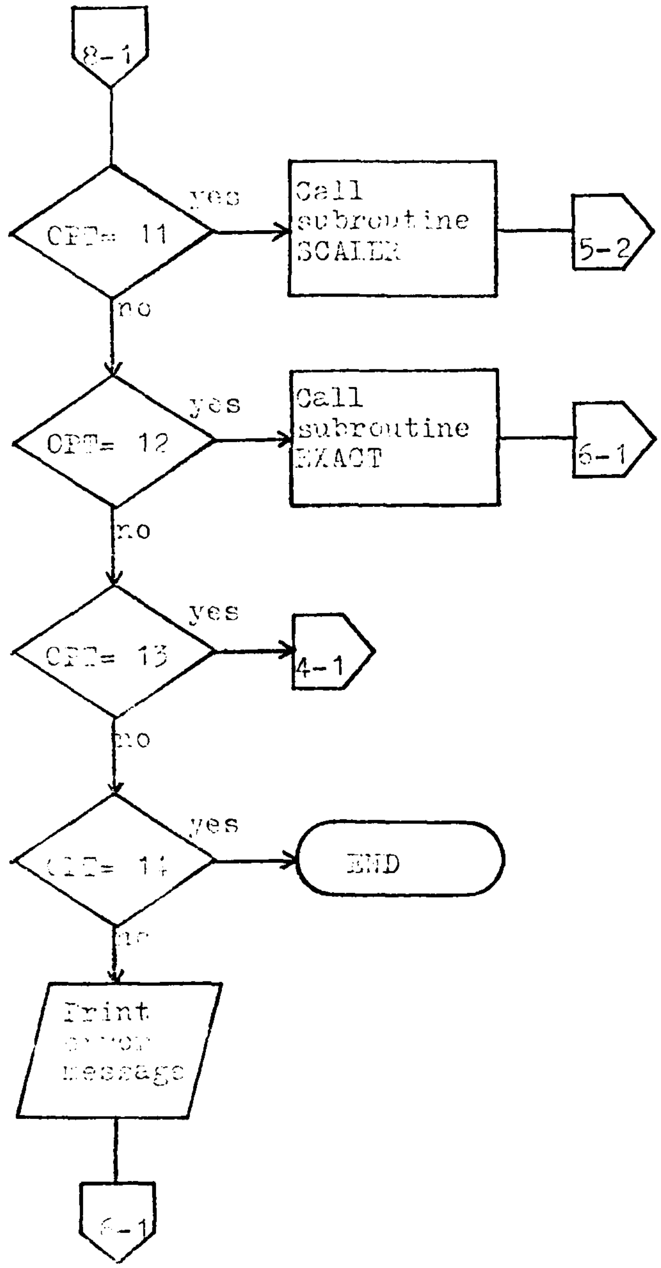



no
yes
Print notice
Check for duplicate signals
any duplicate
yes
Print error message
no
4-1
Compute equi-probable prob
A
A
Call subroutine
Call subroutine
Call subroutine
Call subroutine CMPUTE
CHECK < 0
yes
Print error message
no
Call subroutine PERROR
6-1

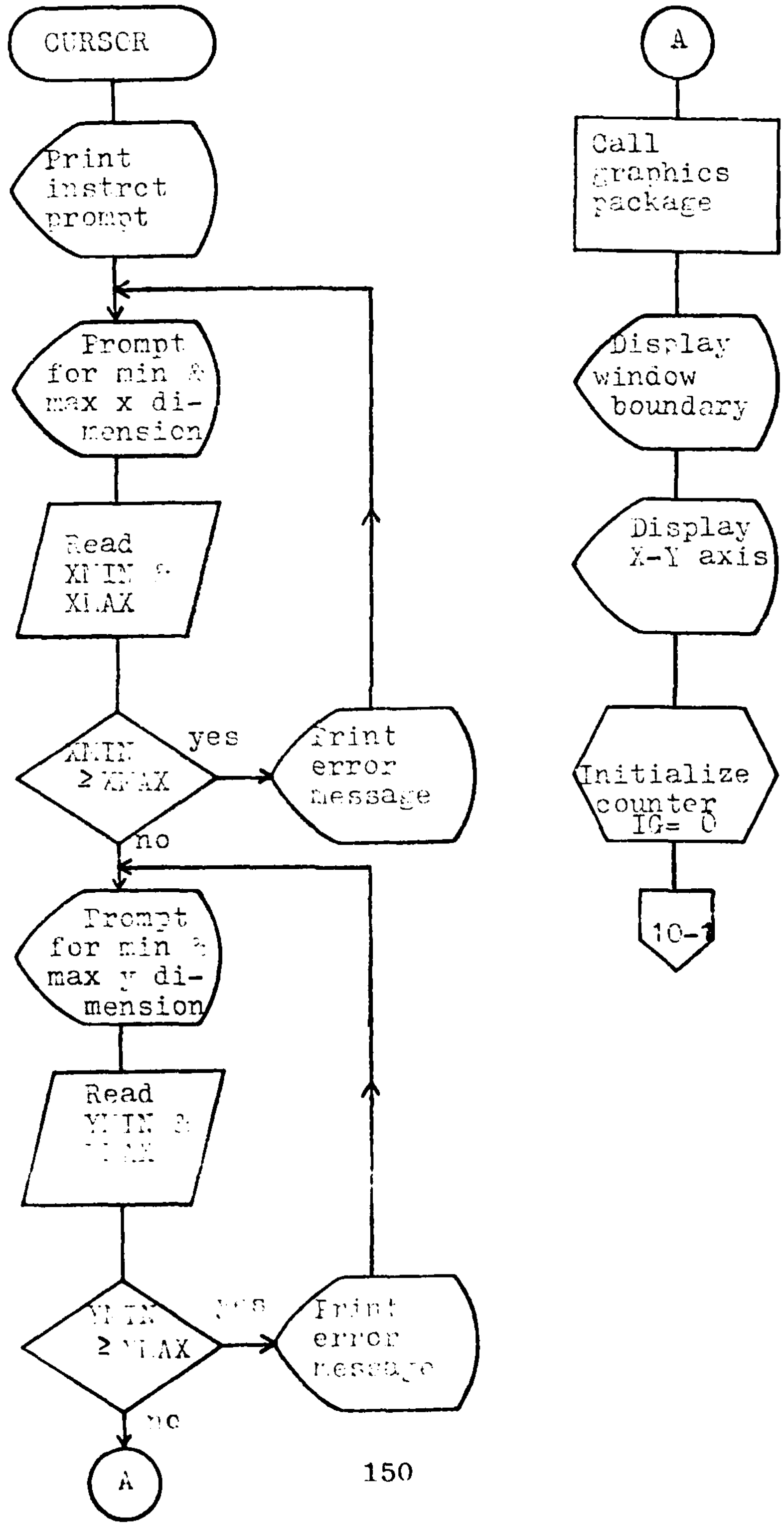

Prompt for OPTION
Read OPT
OPT= 1
yes
Call subroutine TRANS
5-2
no
OPT= 2
yes
Call subroutine ROTATE
5-2
no
OPT= 3
yes
Call subroutine DELETE
5-1
no
yes
Call
no
7-1

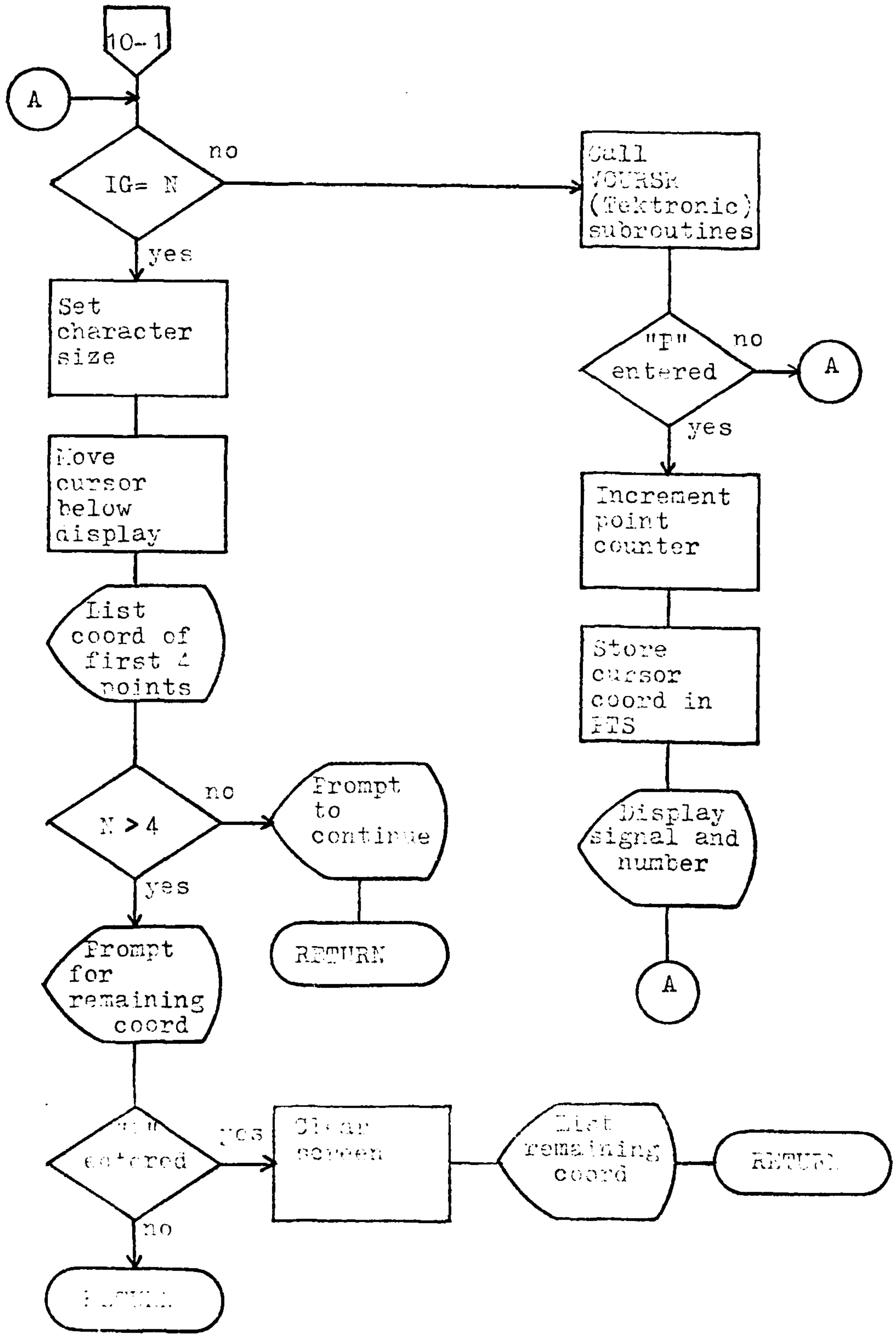


7-1
yes
no
Call subroutine ADDDIR
5-1
Call subroutine
5-2
Call subroutine PLOT
6-1
Call subroutine DECIDE
6-1
Call subroutine OUTPUT
6-1
9

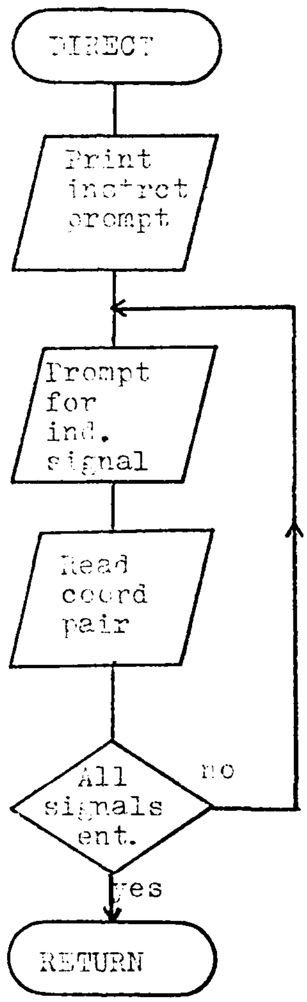

8-1
yes
Call subroutine SCALER
5-2
no
yes
OPT= 12
Call subroutine EXACT
6-1
no
yes
4-1
no
yes
END
Print
message

T. Subroutine CURSCR

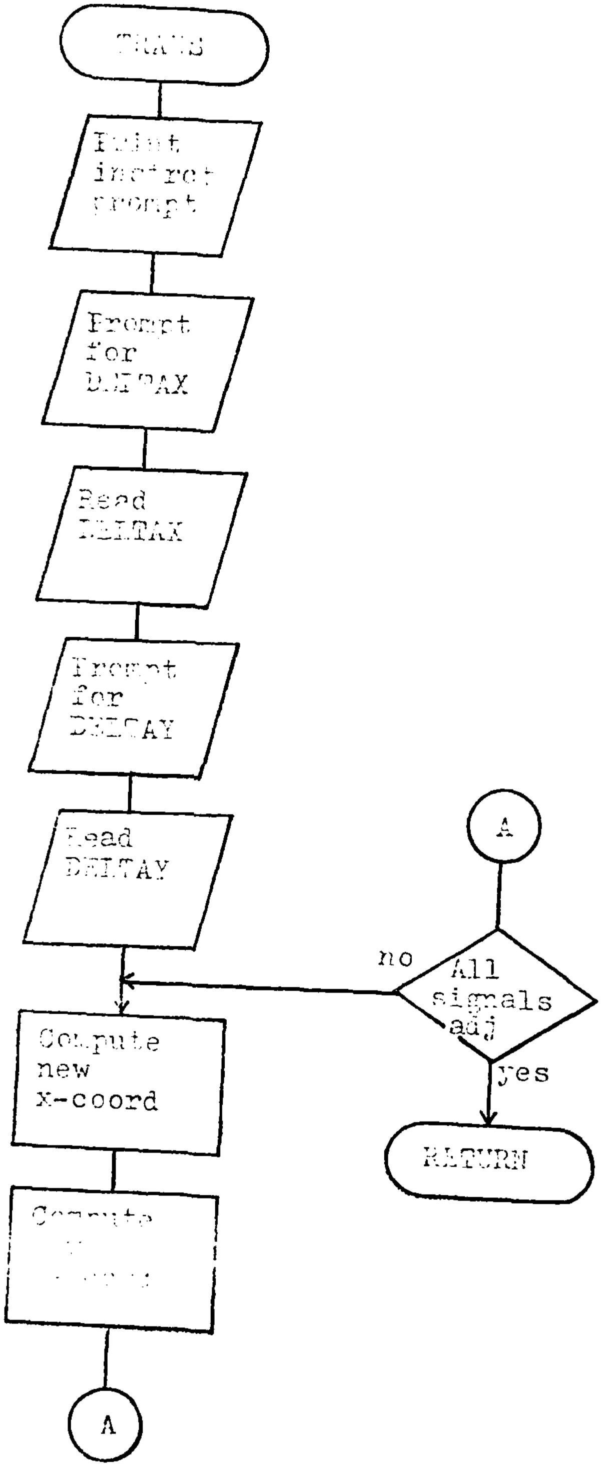



10-1
A
IG= N
no
yes
Call
VCURSR
(Tektronic)
subroutines
Set character size
"P" entered
no
A
yes
Move cursor below display
Increment point counter
List coord of first 4 points
Store cursor coord in
N > 4
no
Prompt to continue
Display signal and number
yes
Prompt for remaining coord
RETURN
A
entered
yes
remaining coord
no

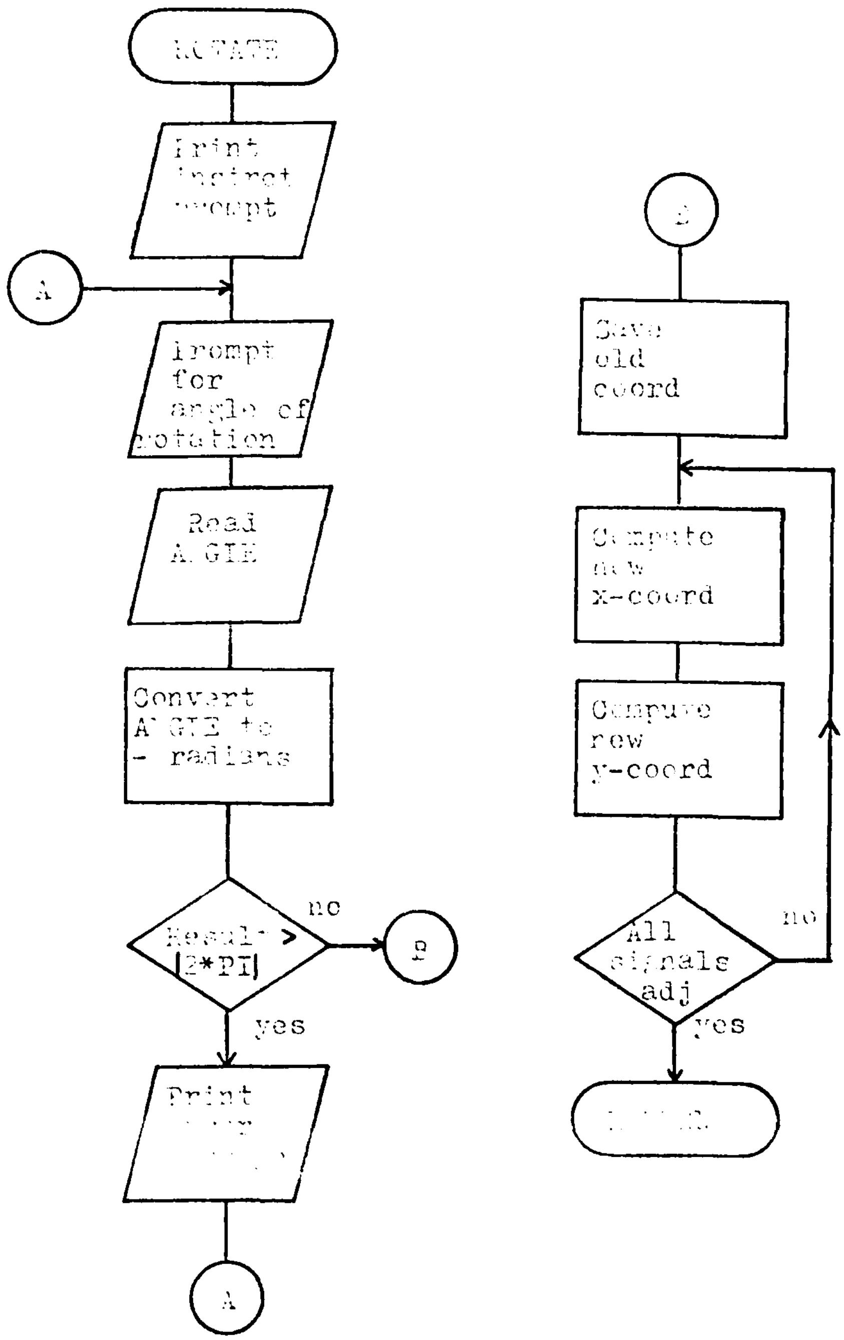
5. Subroutine DIRECT



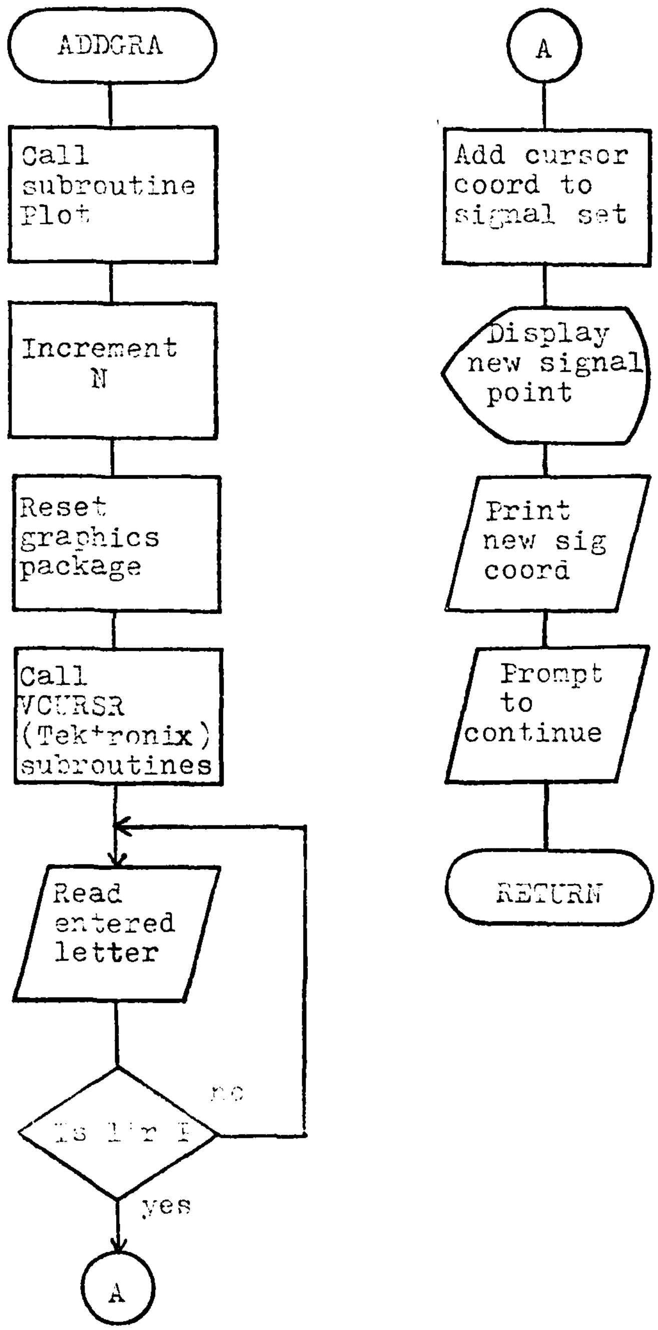
4. Subroutine TRANS



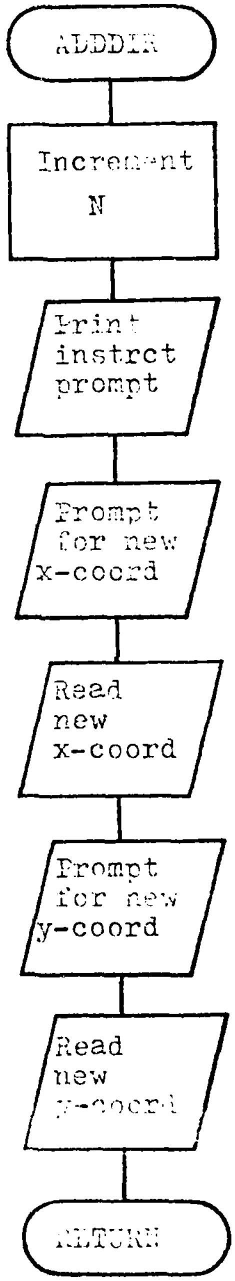
5. Subroutine ROTATE



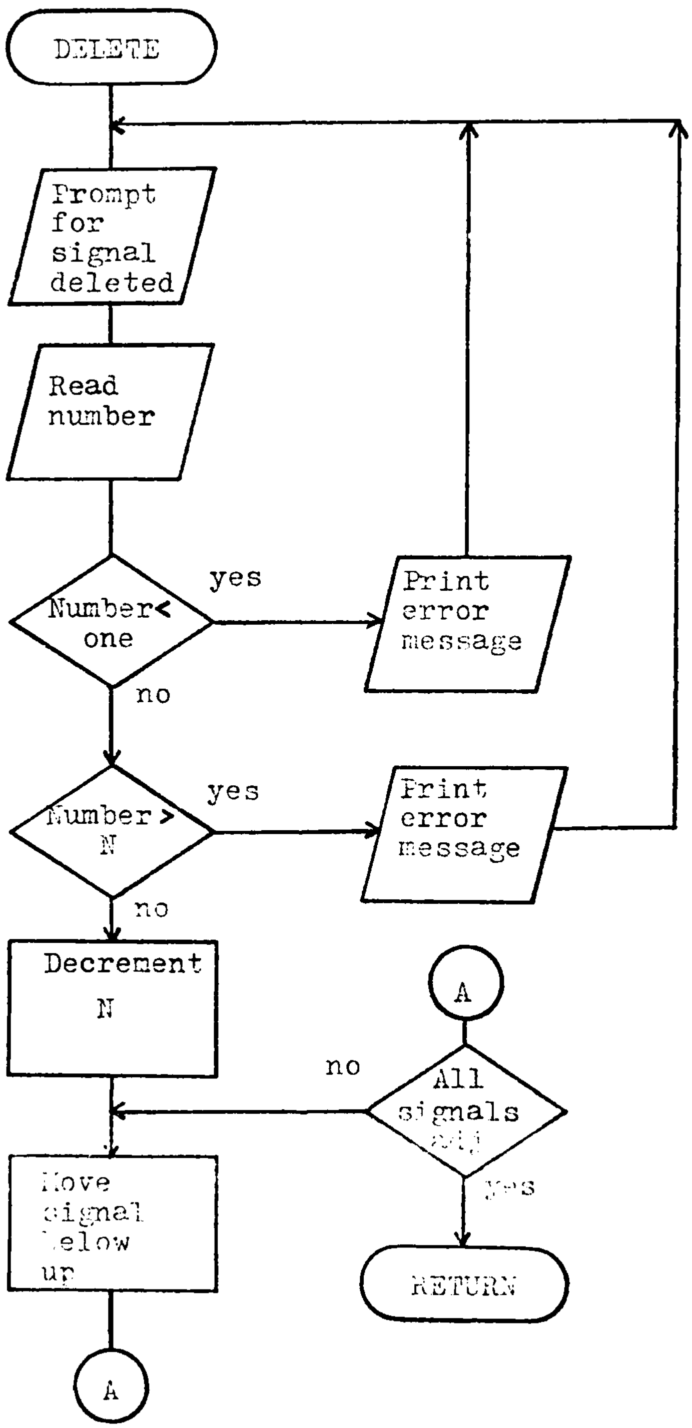
6. Subroutine ADDGRA



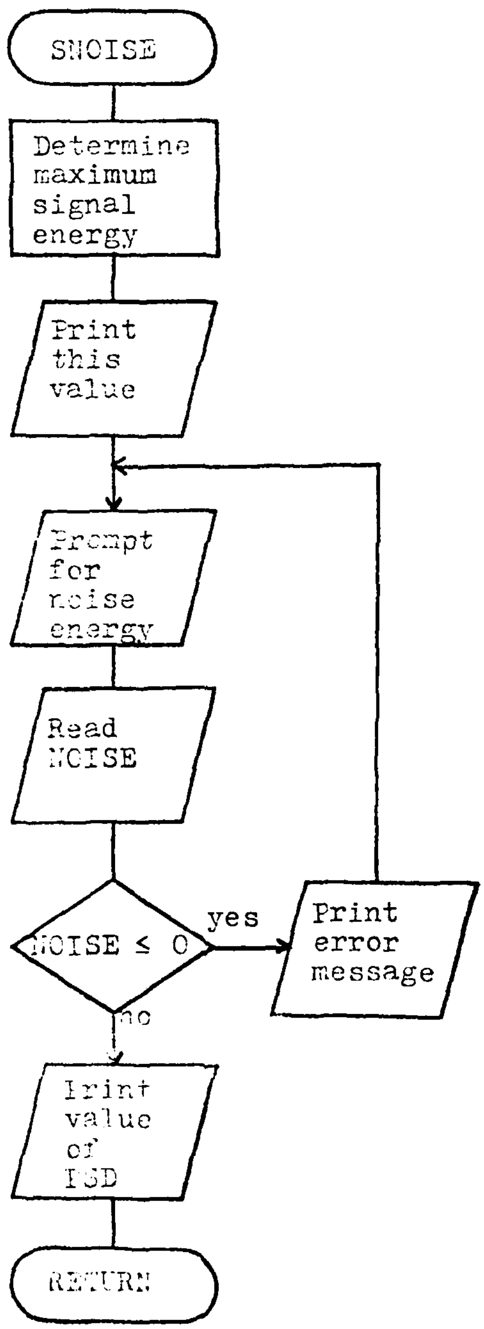
7. Subroutine ADDDIR



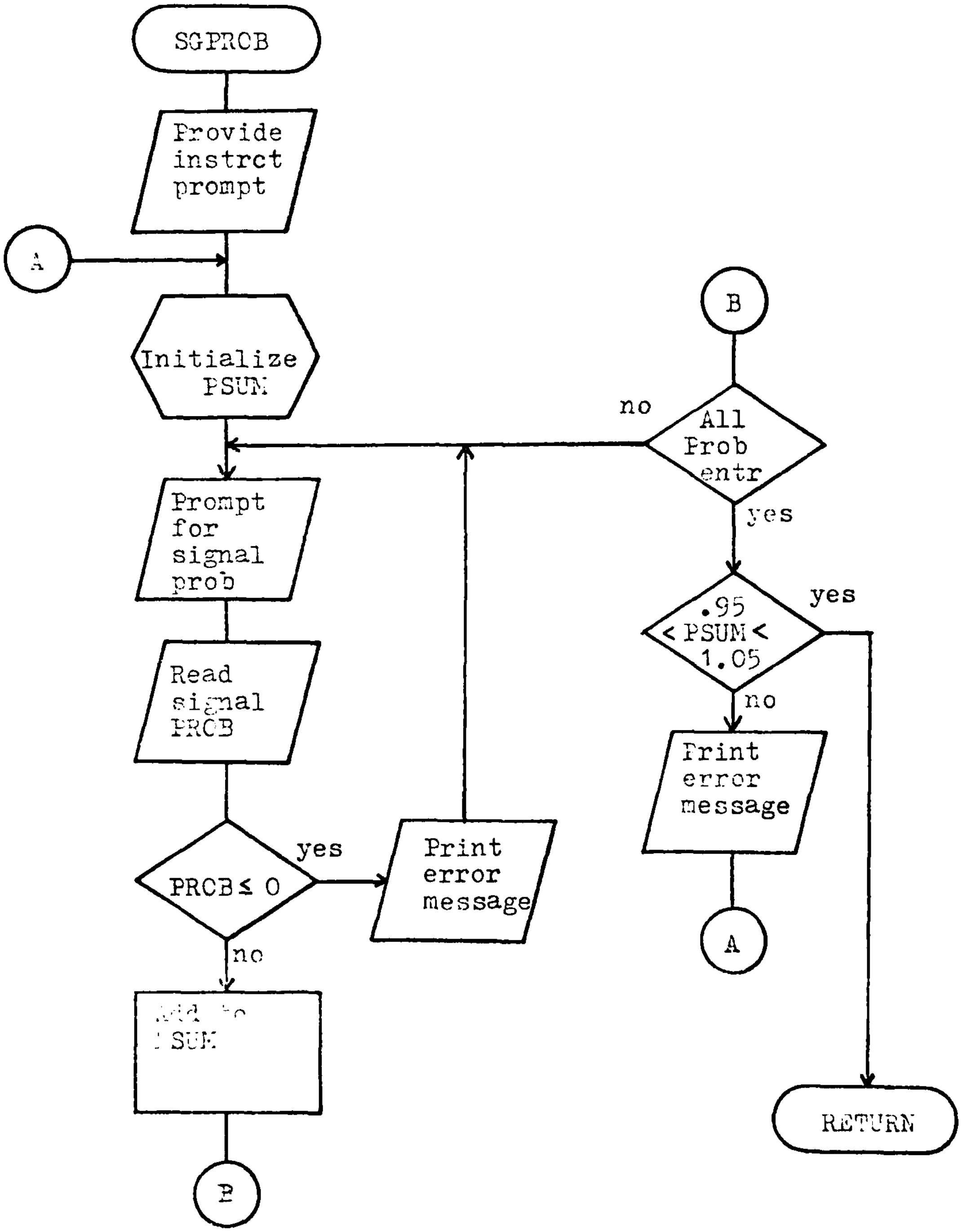
8. Subroutine DELETE



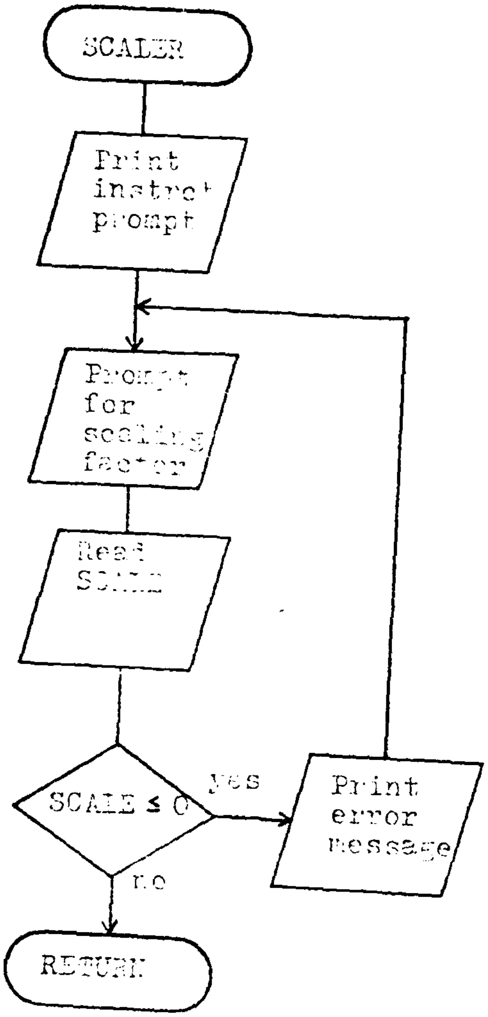
9. Subroutine SNOISE



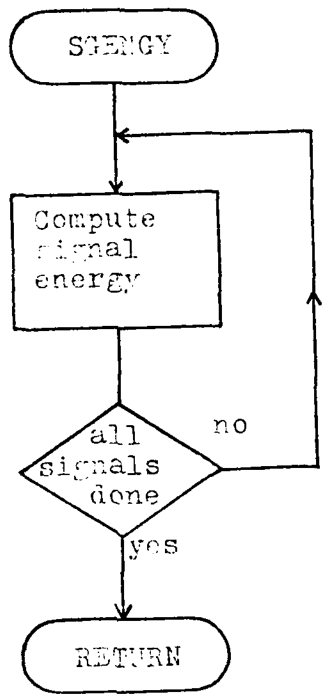
10. Subroutine SGPROB



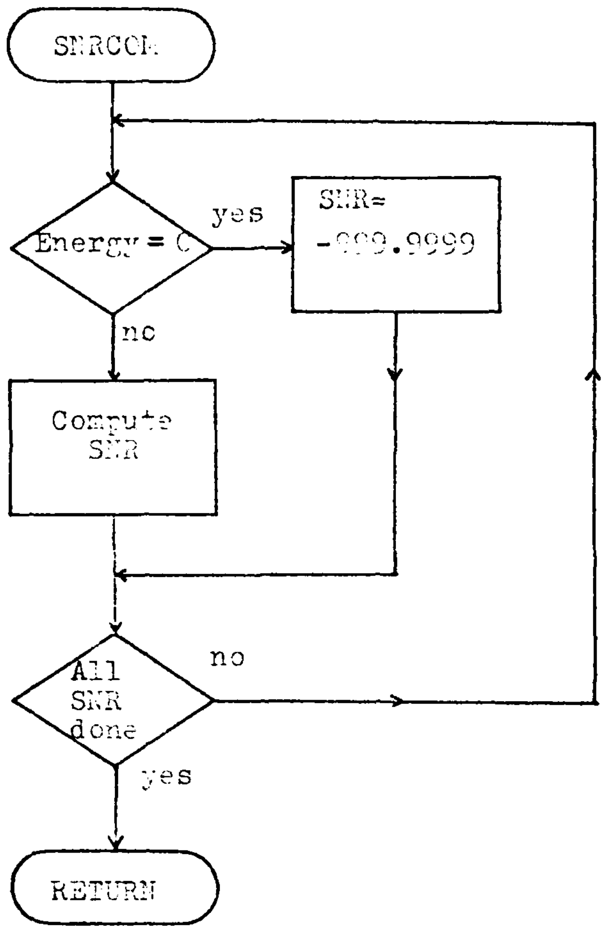
11. Subroutine SCALER



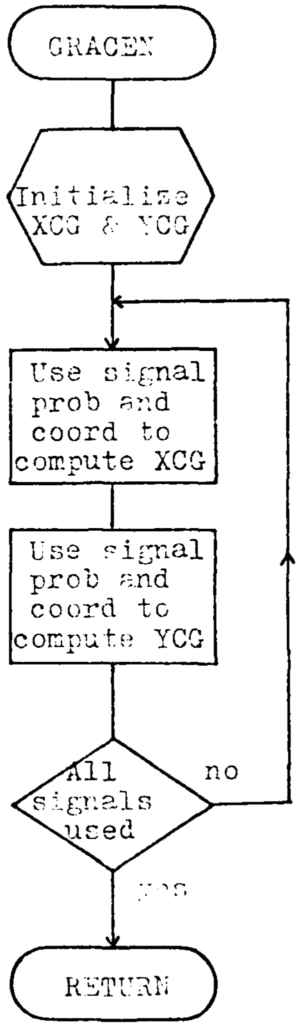
18. Subroutine SGENGY



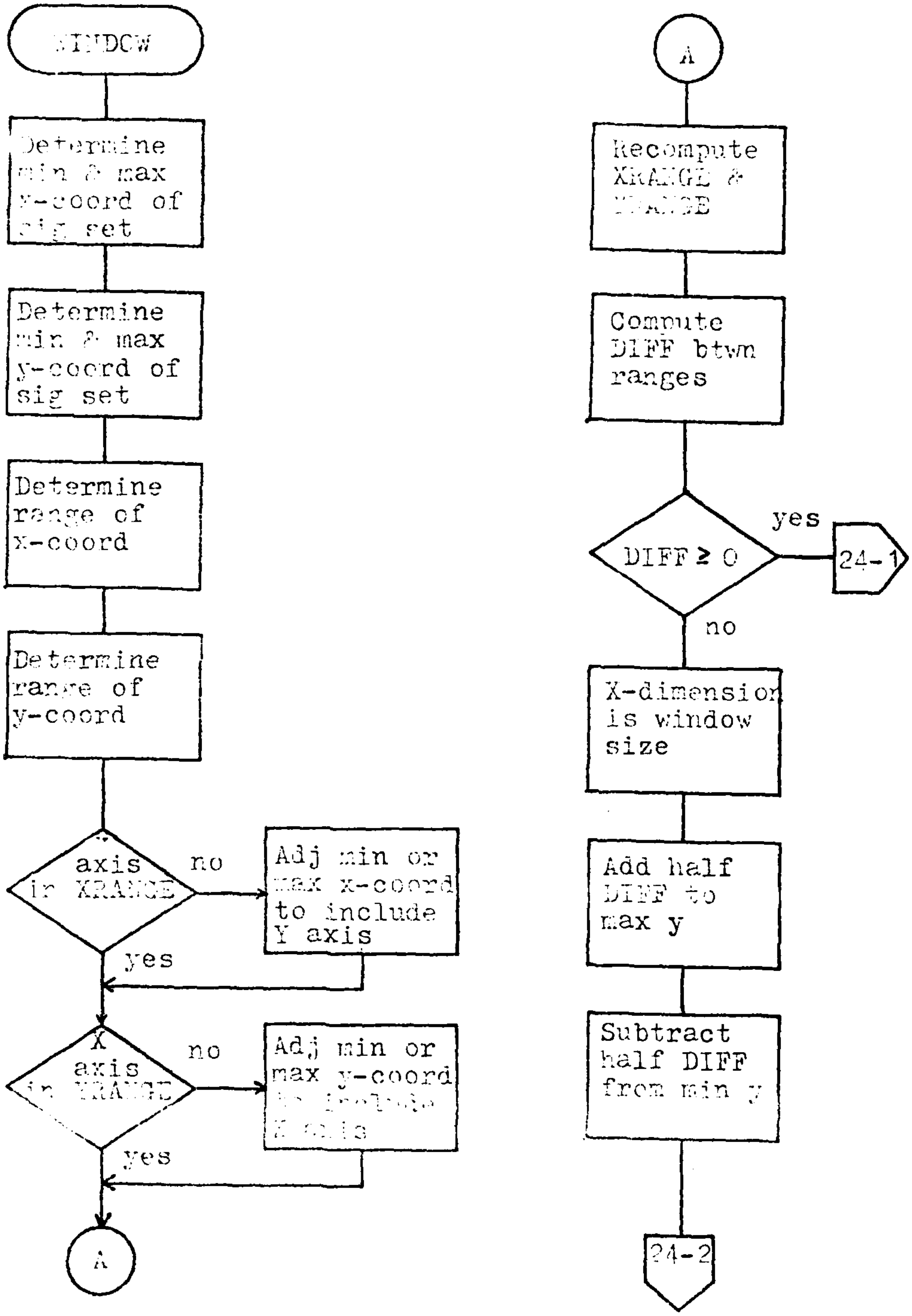
15. Subroutine SNRCOM

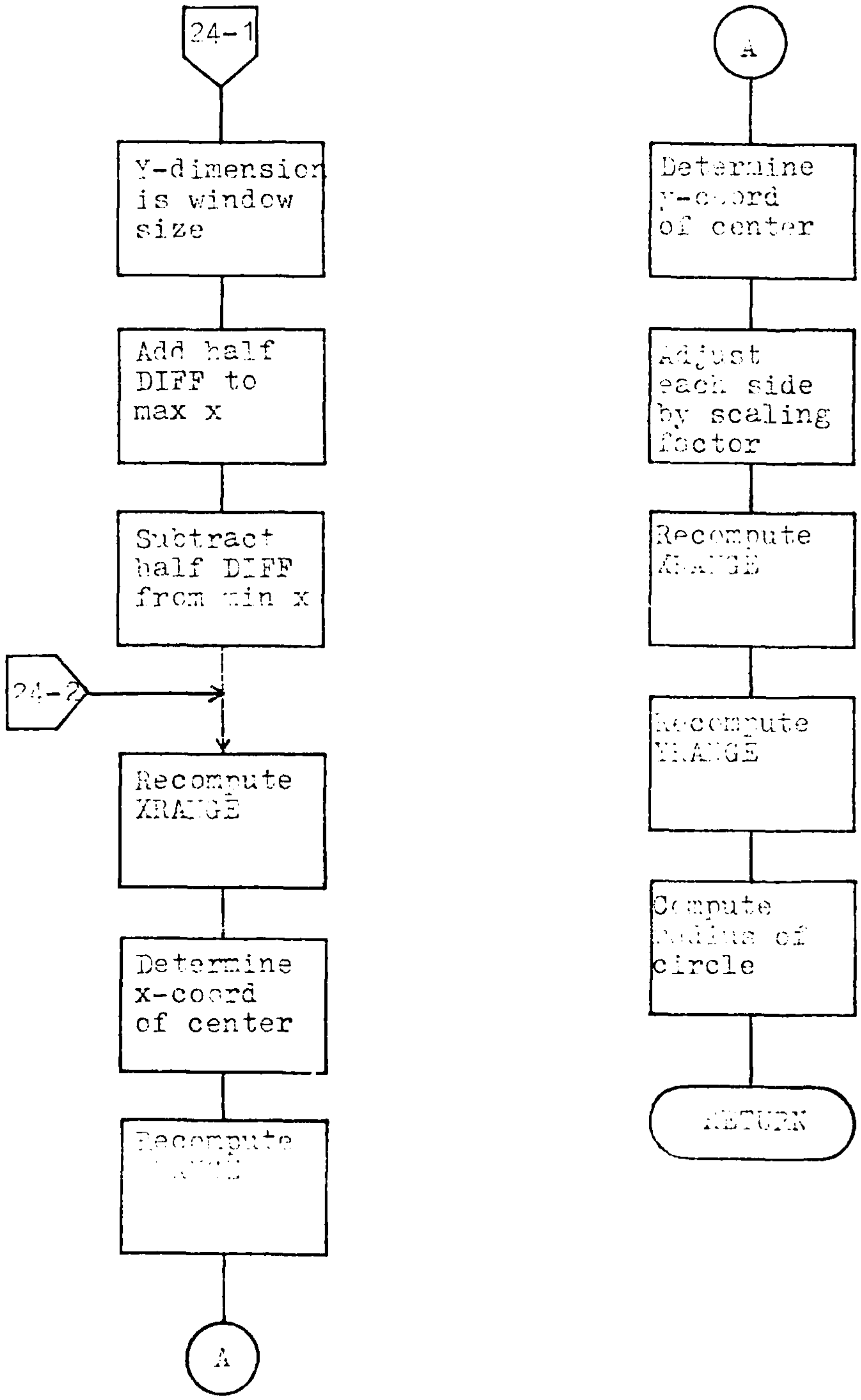


14. Subroutine GRACEN

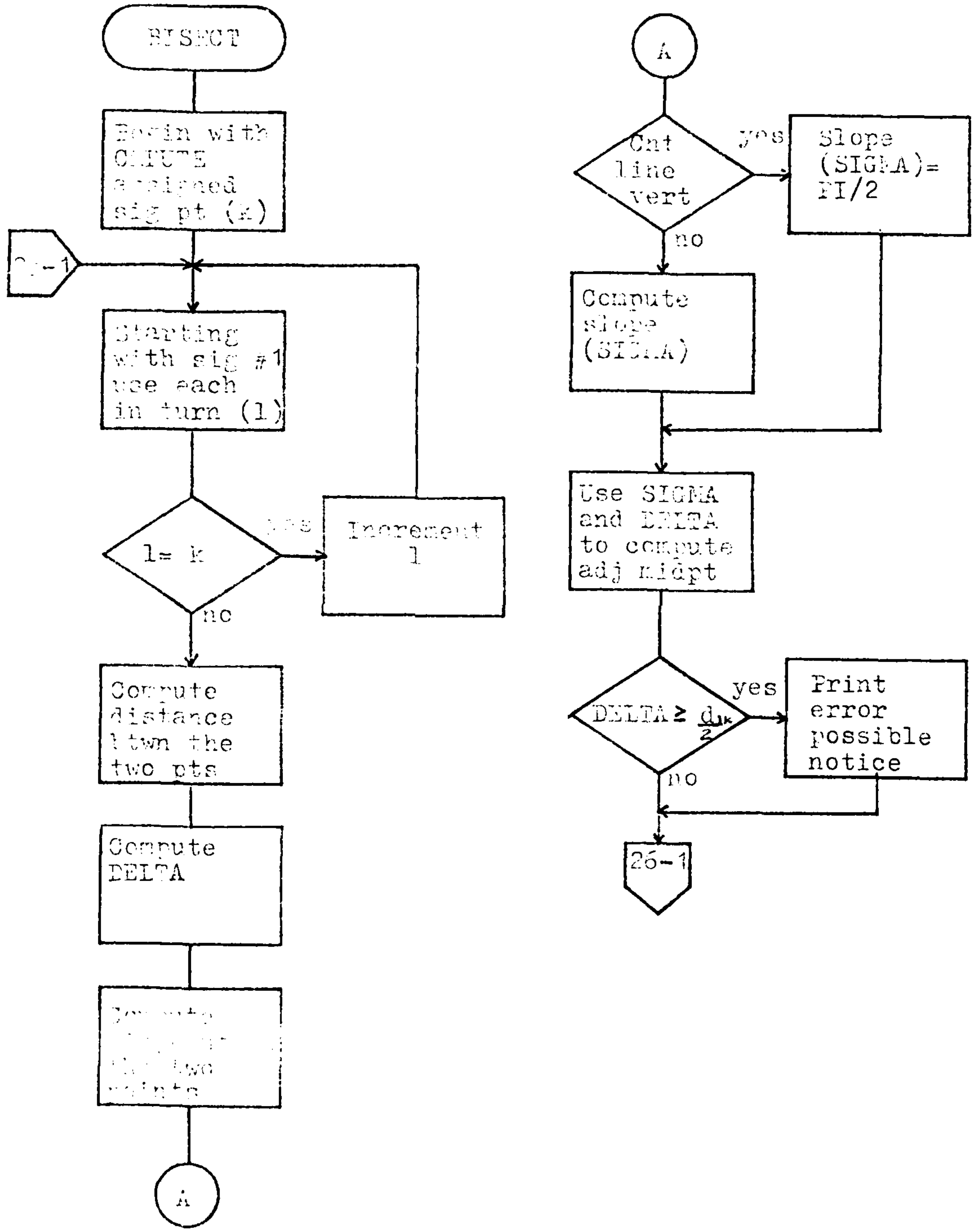


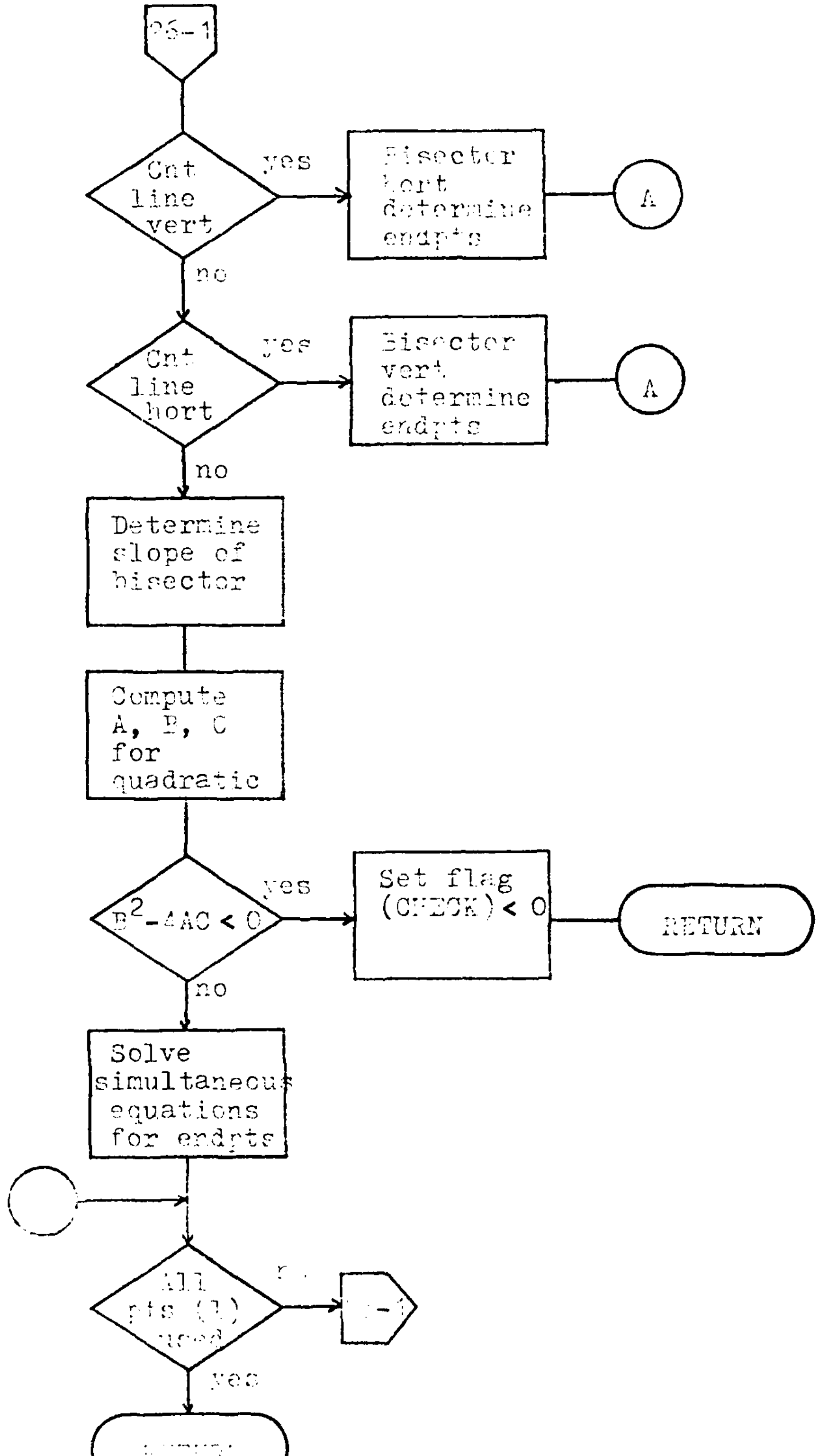
15. Subroutine WINDOW



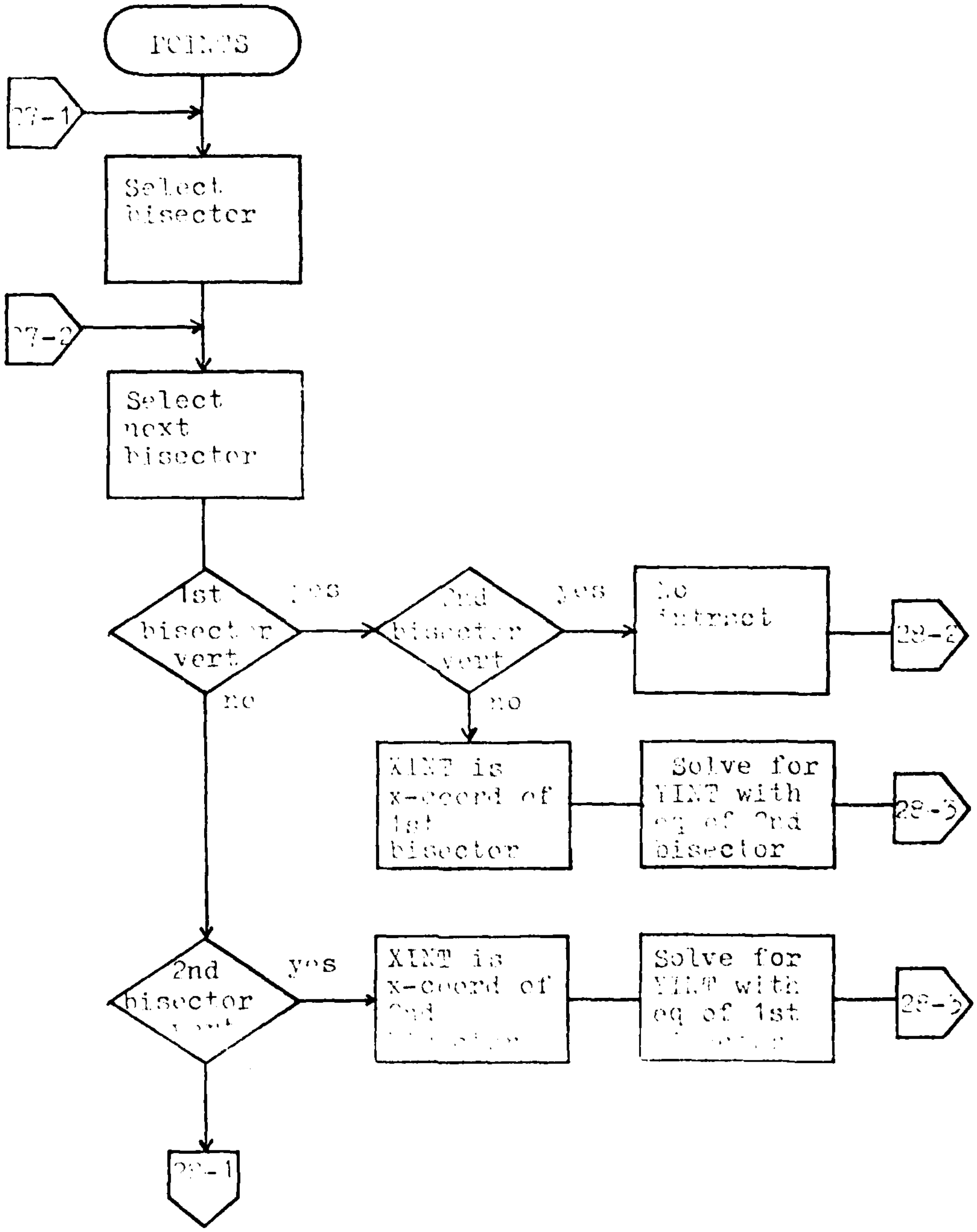

24-1
Y-dimension is window size
Add half DIFF to max x
Subtract half DIFF from min x
24-2
Recompute XRANGE
Determine x-coord of center
Recompute
A
A
Determine y-coord of center
Adjust each side by scaling factor
Recompute XRANGE
Recompute YRANGE
Compute radius of circle
RETURN

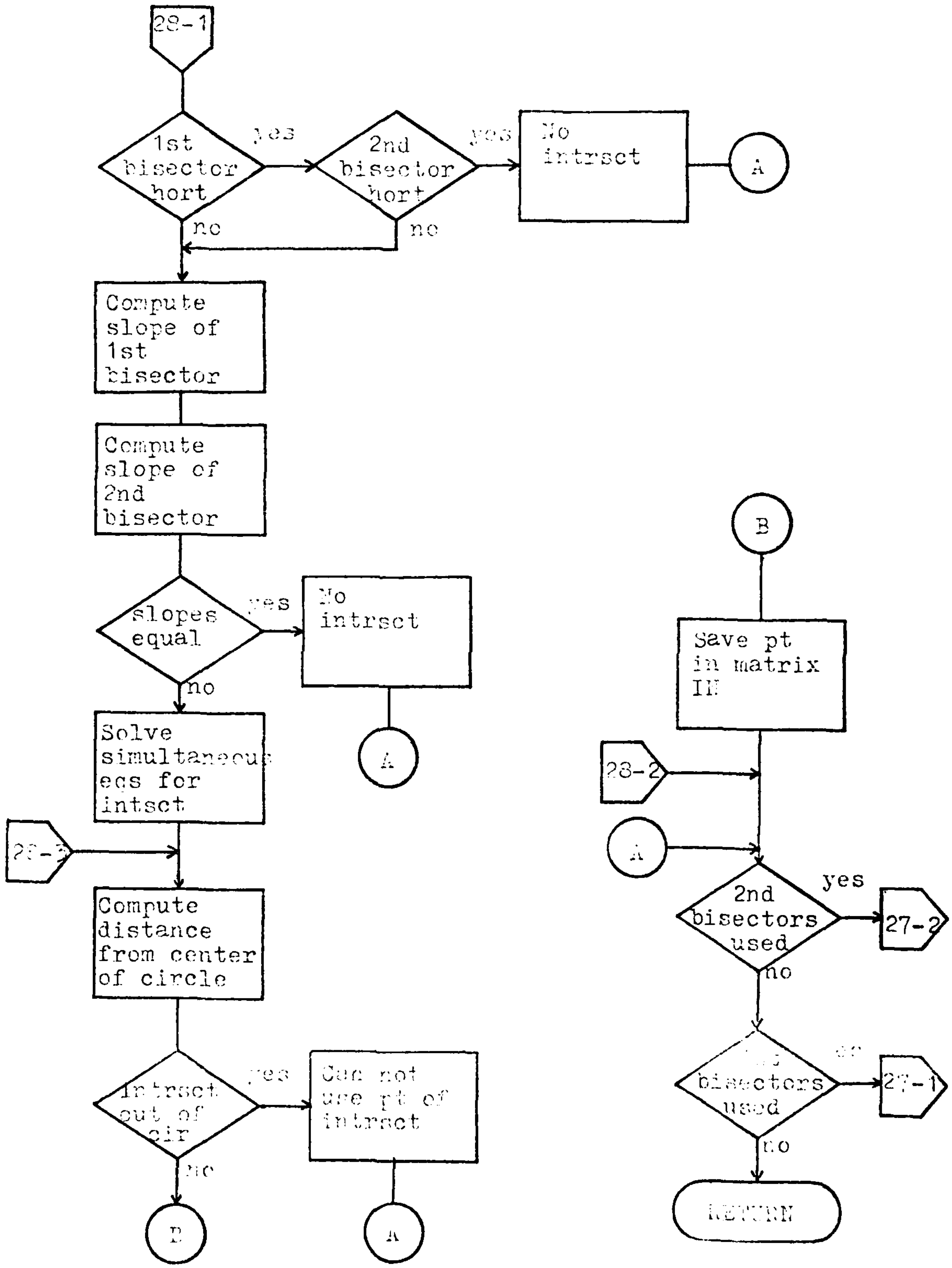
H. Subroutine BISECT



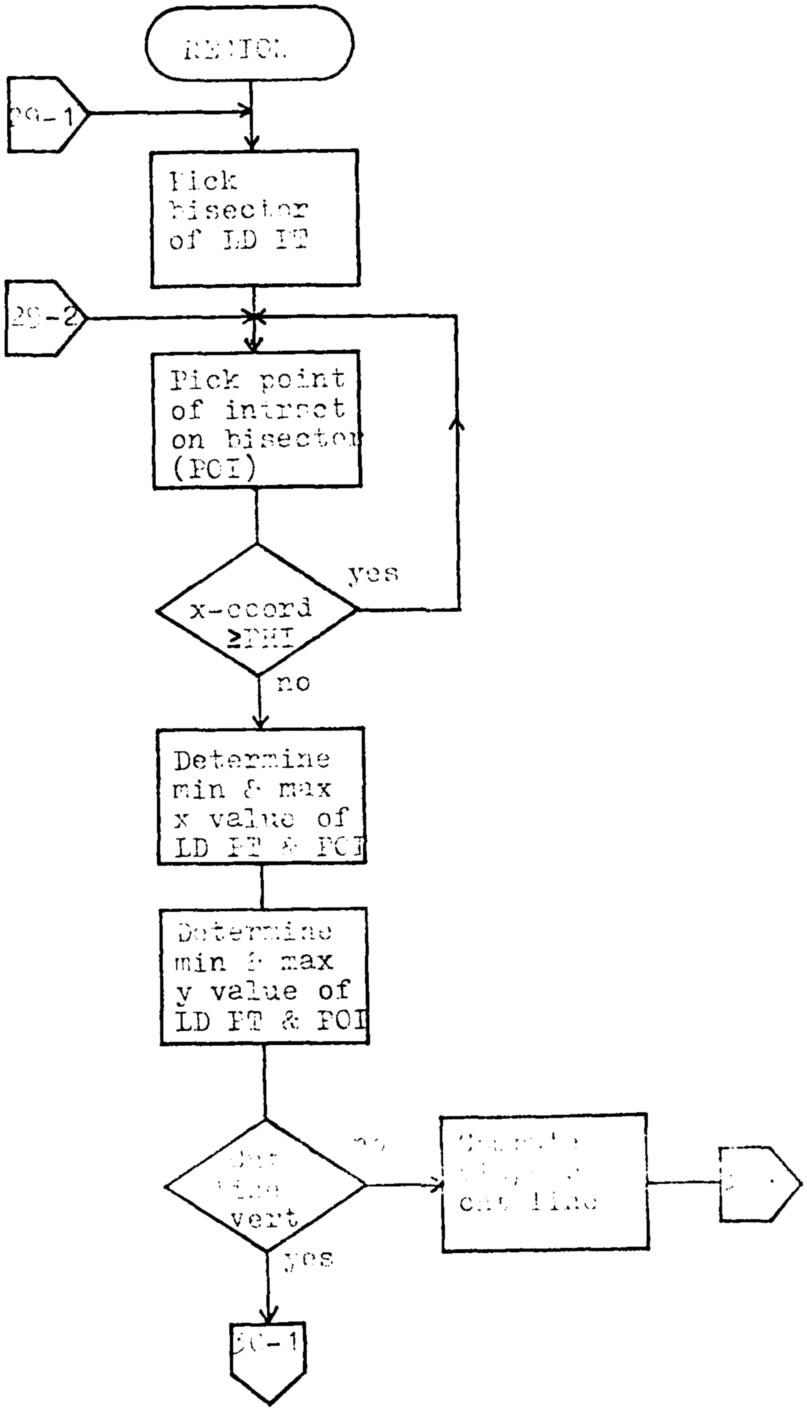

26-1
Cnt line vert
yes
Bisector hort determine endpts
A
no
Cnt line hort
yes
Bisector vert determine endpts
A
no
Determine slope of bisector
Compute A, B, C for quadratic
$B^2-4AC < 0$
yes
Set flag (CHECK) < 0
RETURN
no
Solve simultaneous equations for endpts
All pts (I) used
yes
RETURN

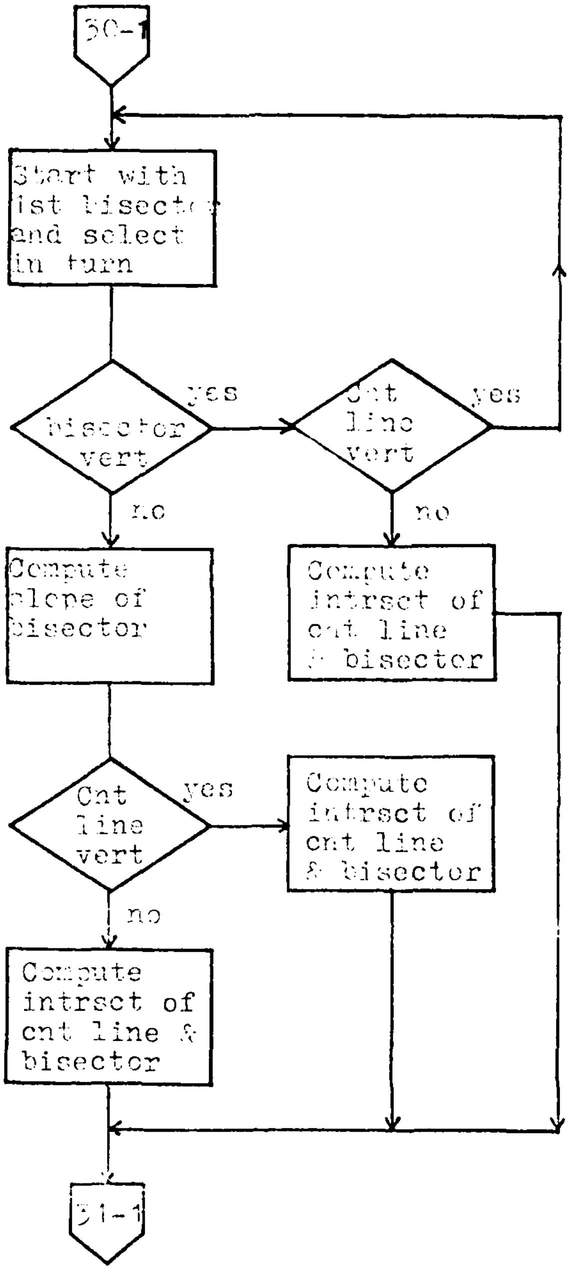
17. Subroutine POINTS

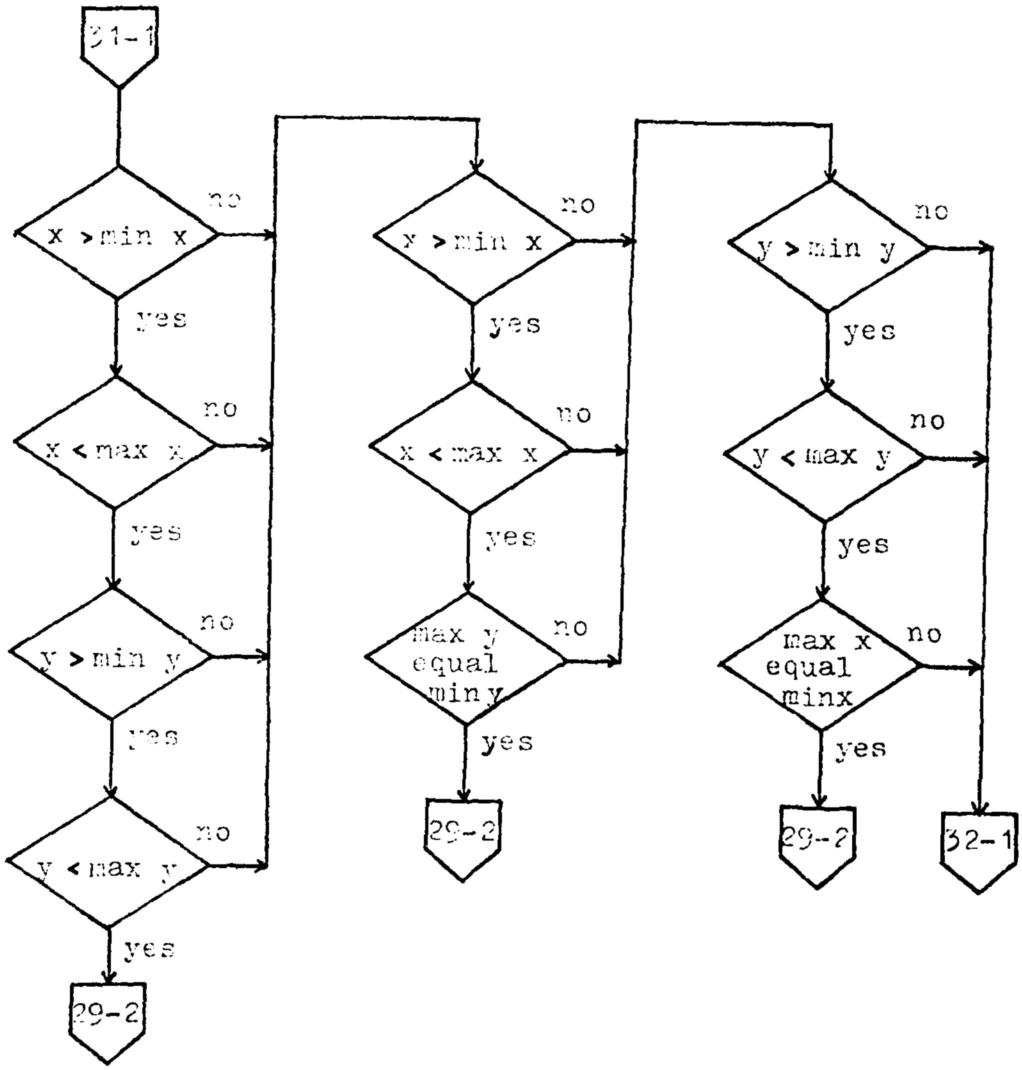


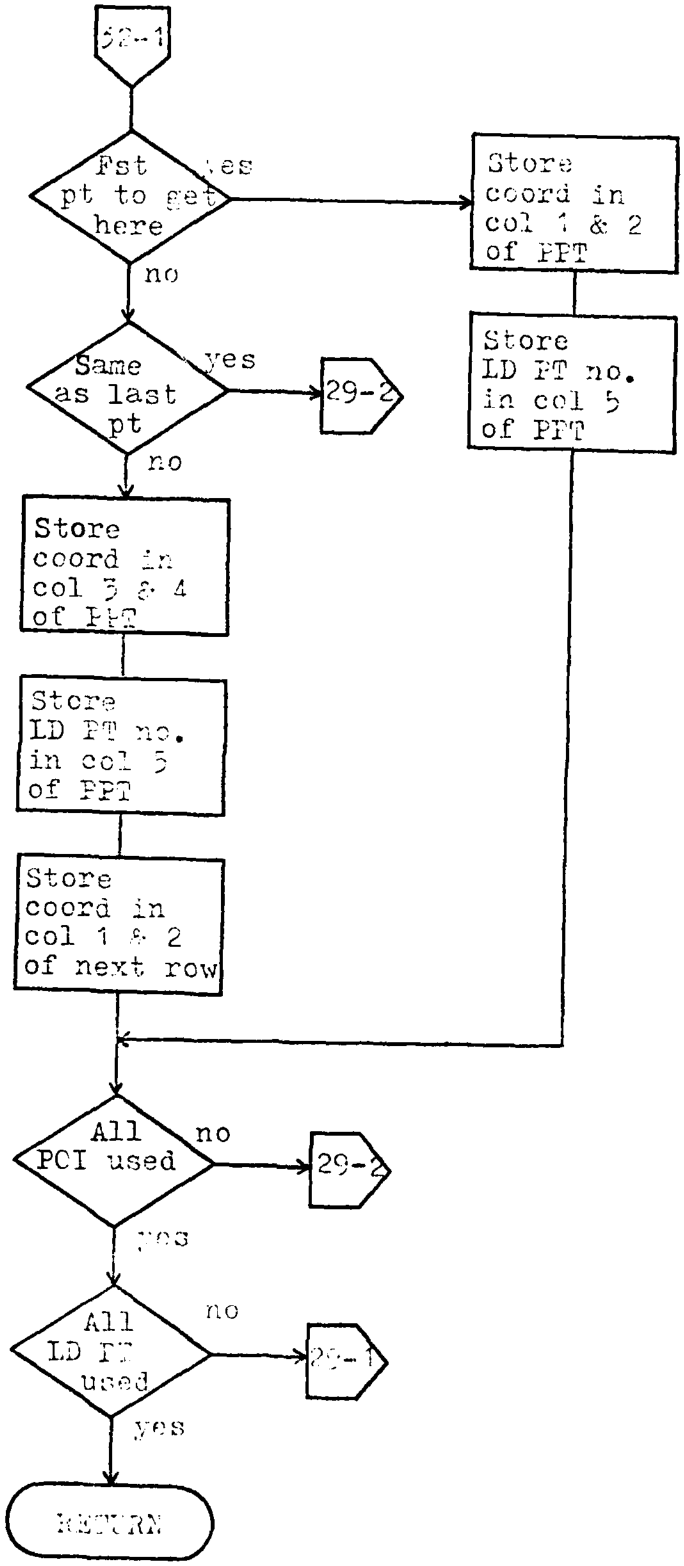

28-1
1st bisector hort
yes
2nd bisector hort
yes
No intrsct
A
no
no
Compute slope of 1st bisector
Compute slope of 2nd bisector
slopes equal
yes
No intrsct
A
no
Solve simultaneous eqs for intsct
Compute distance from center of circle
yes
no
A
B
Save pt in matrix IN
28-2
A
2nd bisectors used
yes
27-2
no
bisectors used
27-1
no

18. Subroutine [illegible]



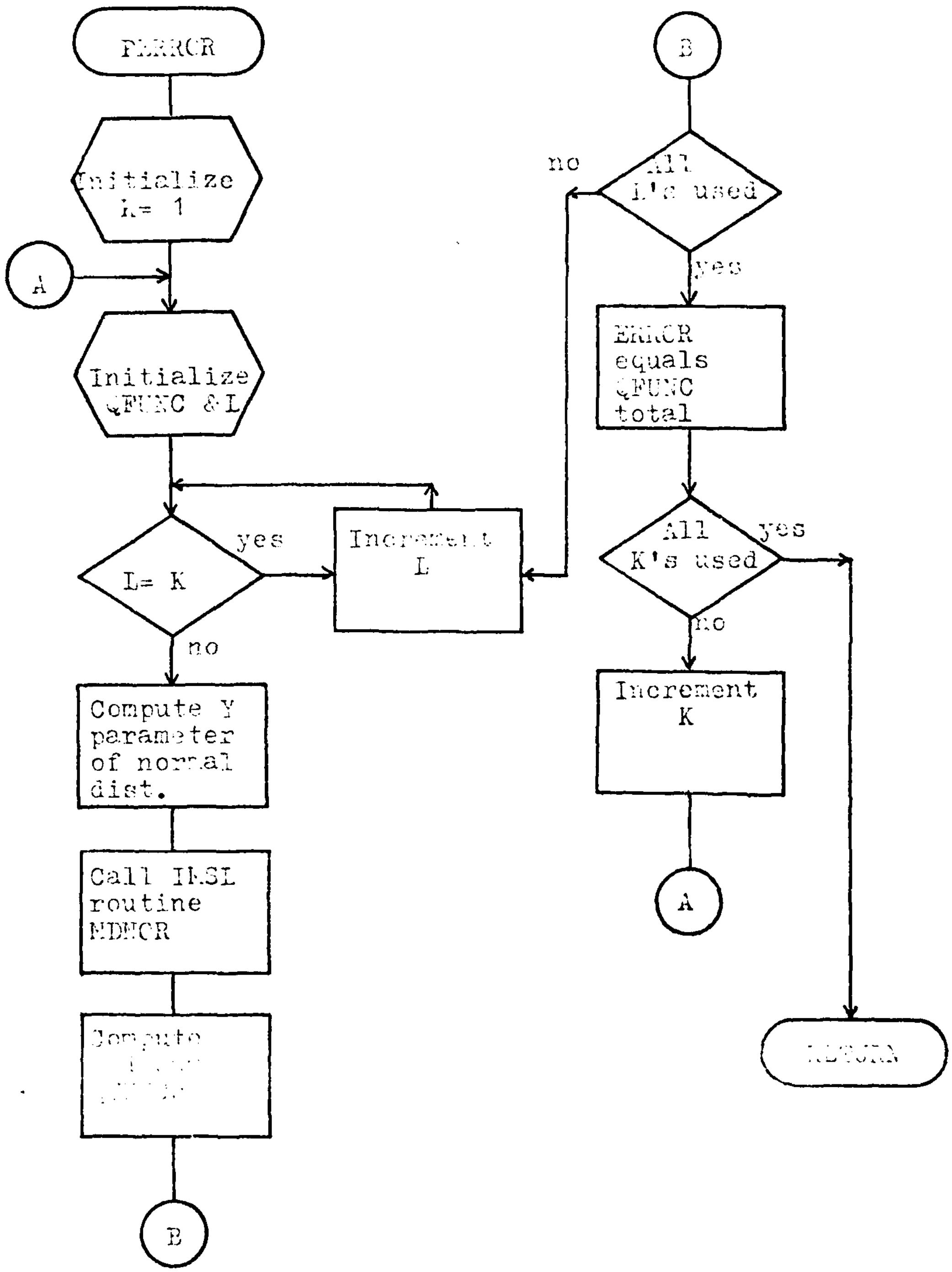

30-1
Start with
1st bisector
and select
in turn
bisector
vert
yes
Cnt
line
vert
yes
no
no
Compute
slope of
bisector
Compute
intrsct of
cnt line
& bisector
Cnt
line
vert
yes
Compute
intrsct of
cnt line
& bisector
no
Compute
intrsct of
cnt line &
bisector
31-1


31-1
x > min x
no
yes
x < max x
y > min y
y < max y
29-2
max y equal min y
y > min y
y < max y
max x equal minx
32-1

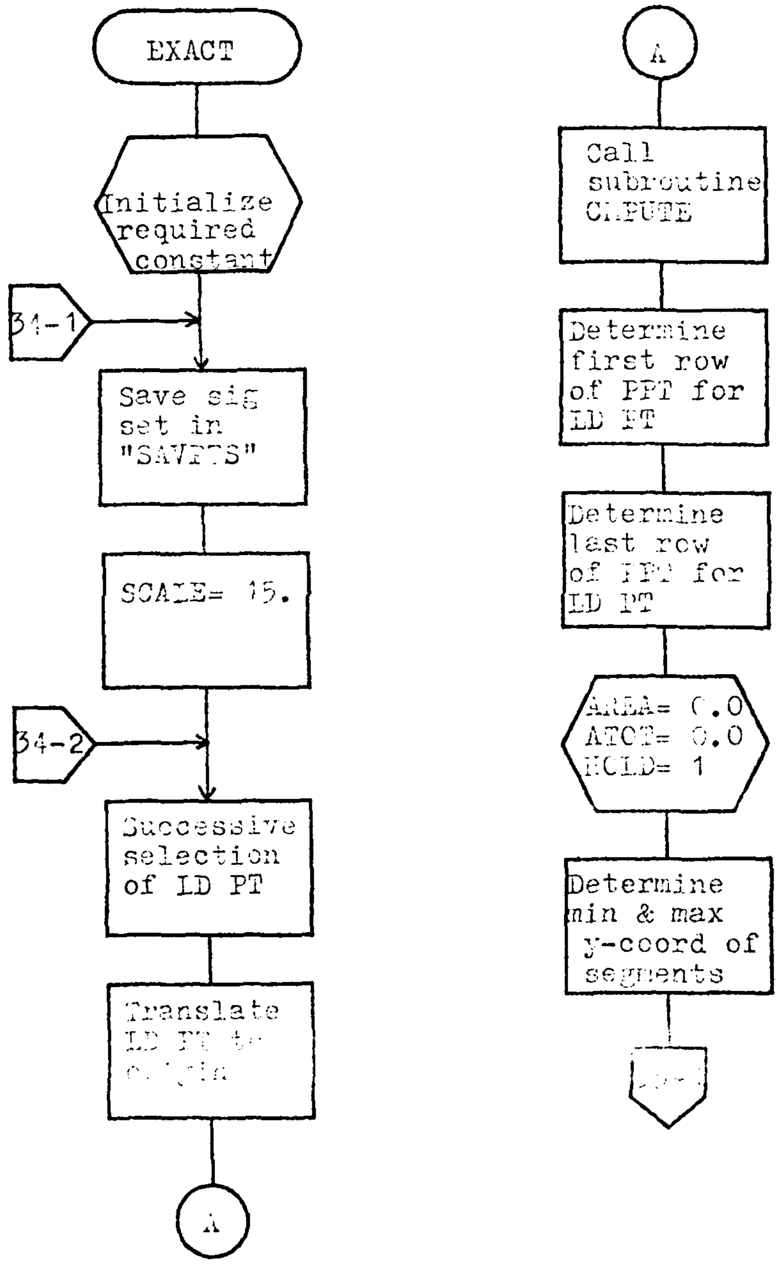


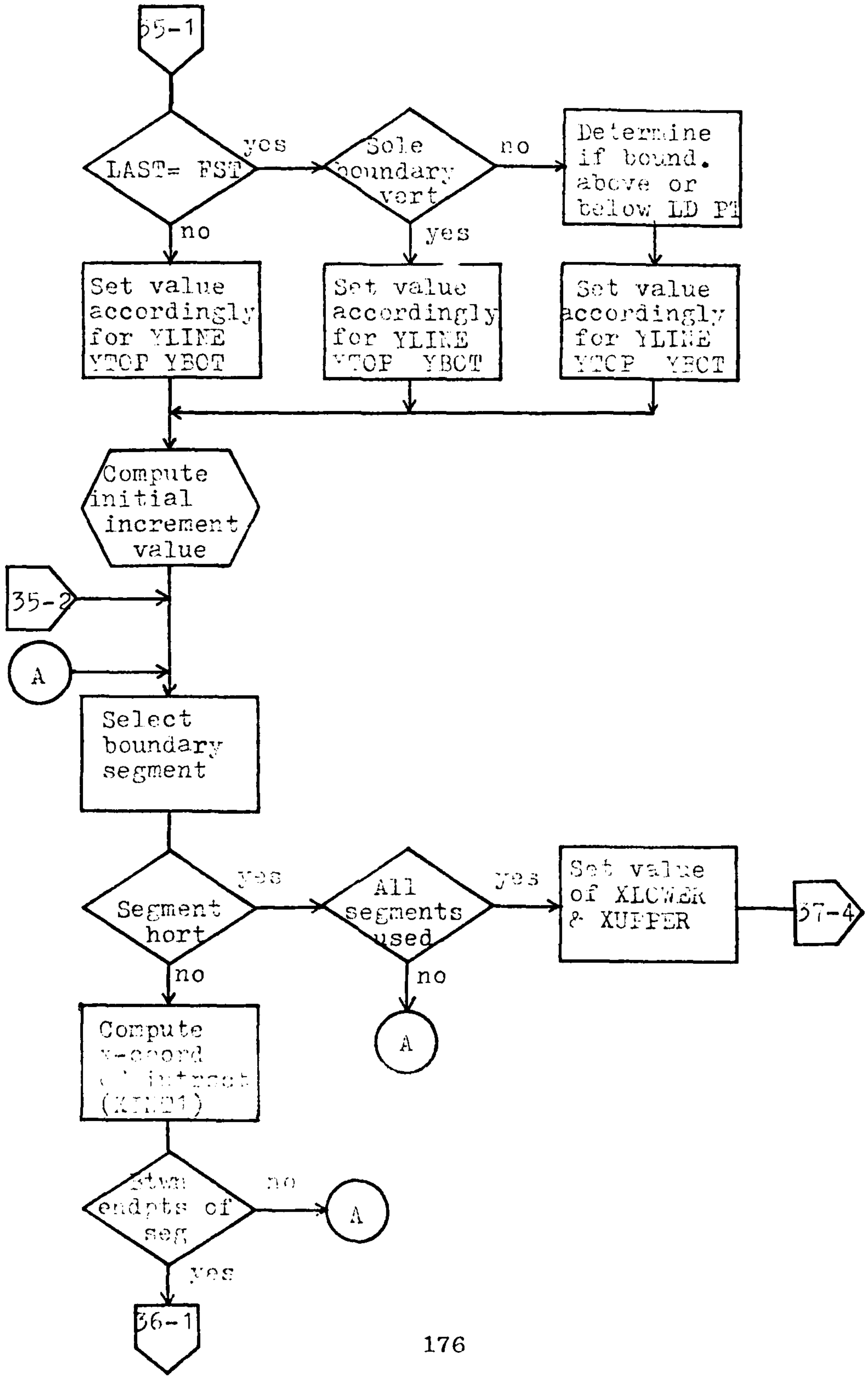
52-1
Fst pt to get here
yes
Store coord in col 1 & 2 of PPT
Store LD PT no. in col 5 of PPT
no
Same as last pt
yes
29-2
no
Store coord in col 3 & 4 of PPT
Store LD PT no. in col 5 of PPT
Store coord in col 1 & 2 of next row
All PCI used
no
29-2
yes
All LD PT used
no
29-1
yes
RETURN

1[illegible]. Subroutine FERROR

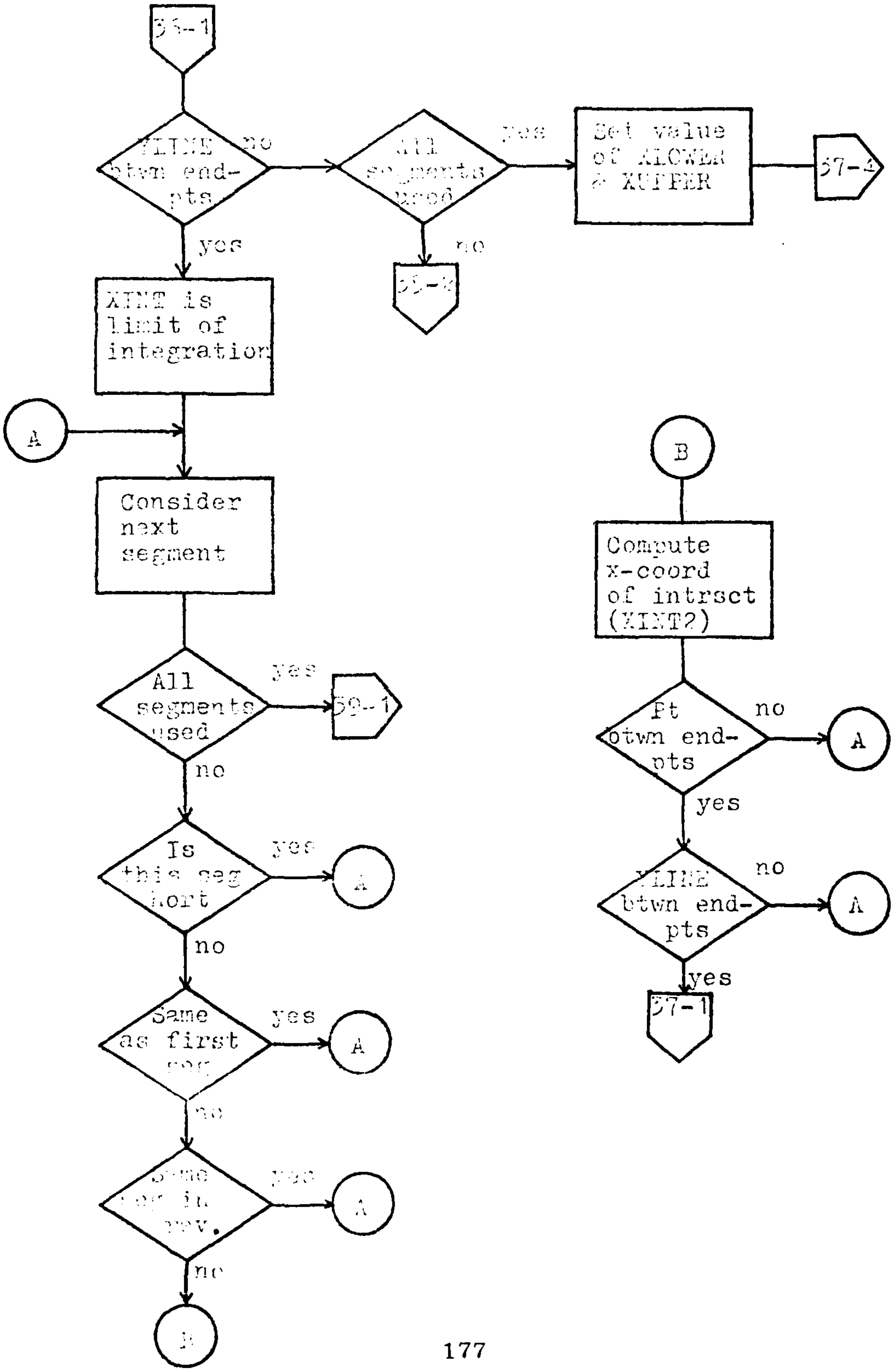


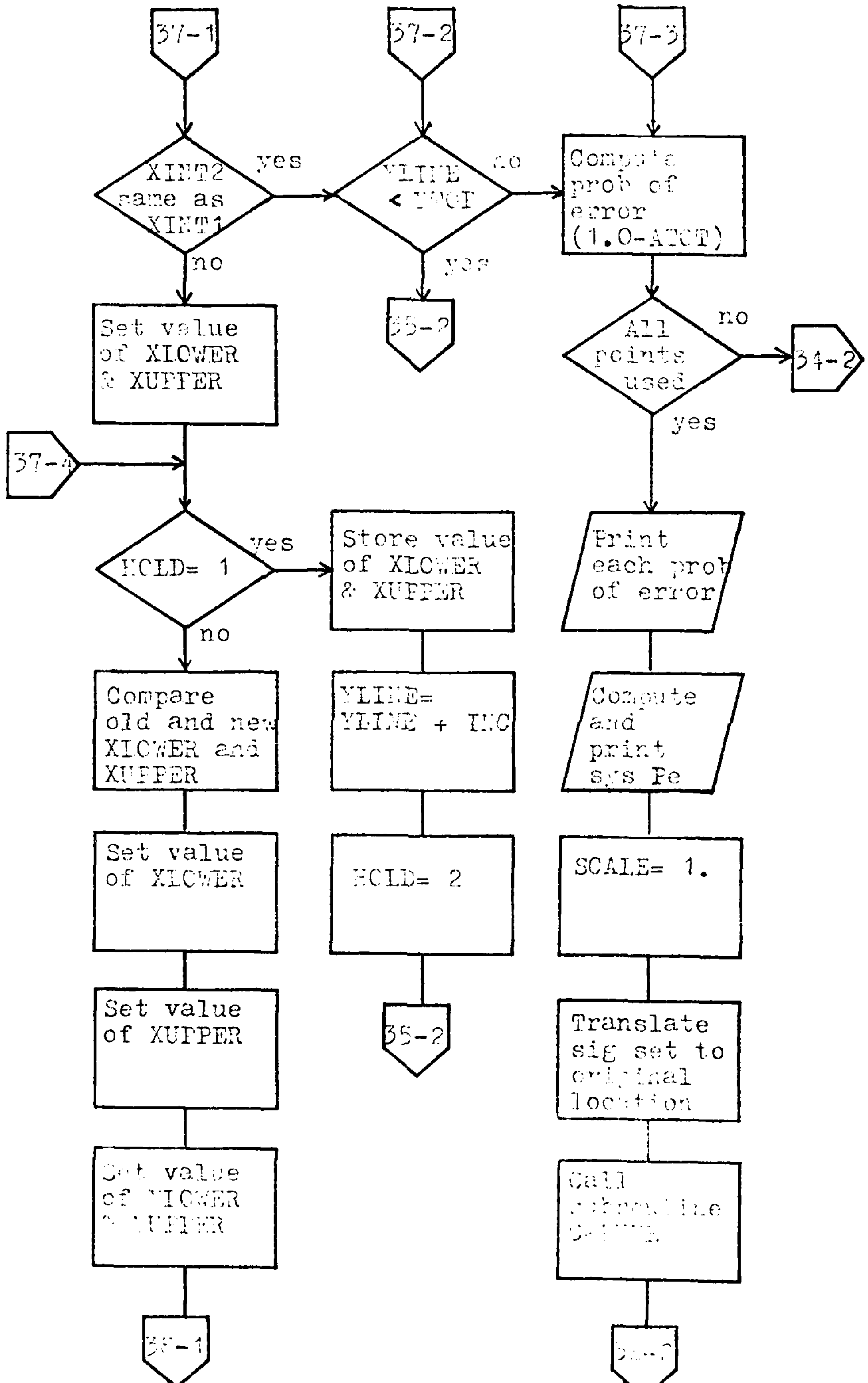
20. Subroutine EXACT



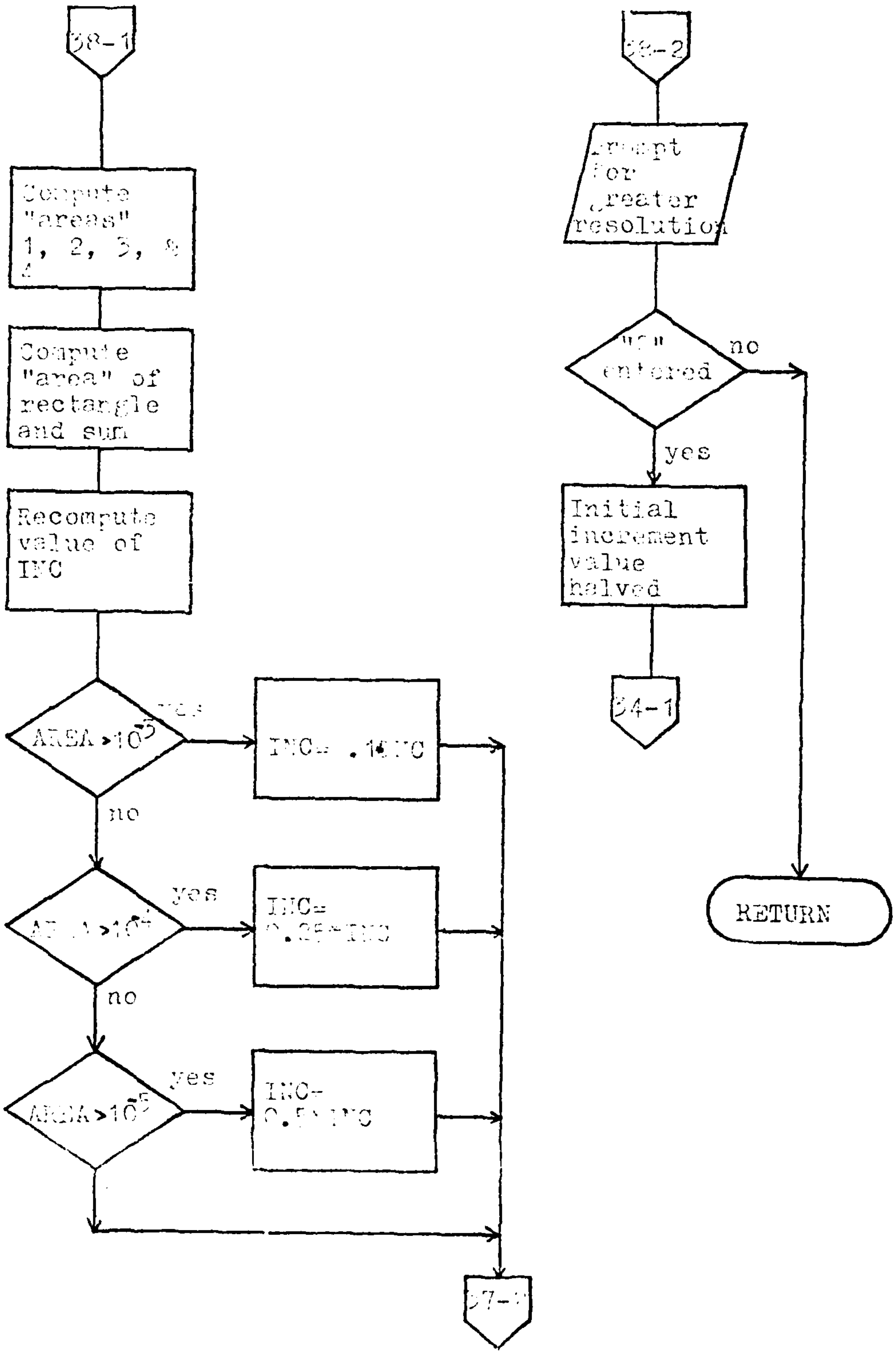


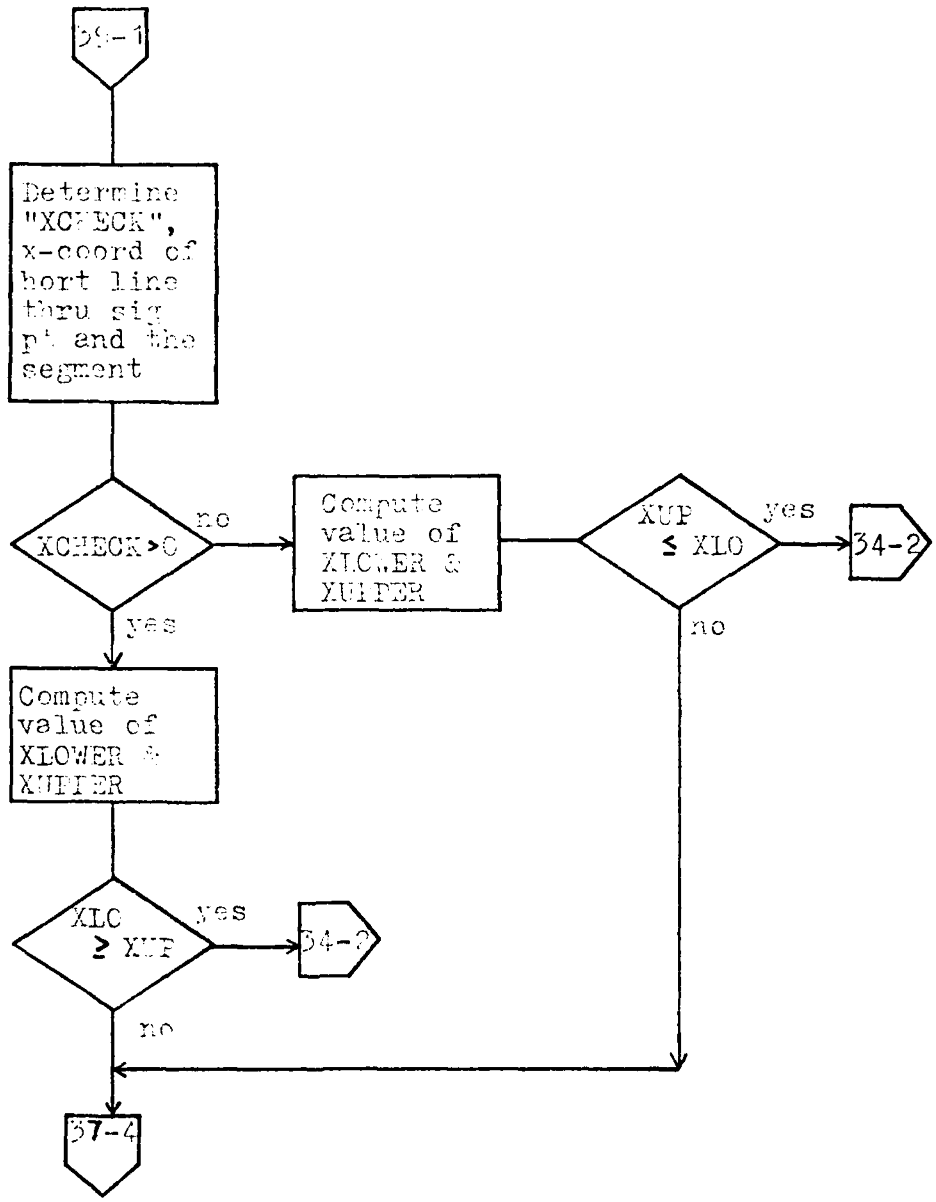
35-1
LAST= FST
yes
no
Sole boundary vert
no
yes
Determine if bound. above or below LD PT
Set value accordingly for YLINE YTOP YBOT
Set value accordingly for YLINE YTOP YBOT
Set value accordingly for YLINE YTOP YBOT
Compute initial increment value
35-2
A
Select boundary segment
Segment hort
yes
no
All segments used
yes
no
Set value of XLOWER & XUPPER
37-4
A
Compute
(XINT1)
endpts of seg
no
yes
A
36-1


yes
no
Set value
& XUPPER
57-4
yes
no
XINT is
limit of
integration
A
Consider
next
segment
All
segments
used
yes
59-1
no
Is
yes
A
no
Same
as first
yes
A
yes
A
B
B
Compute
x-coord
of intrsct
(XINT2)
Pt
btwn end-
pts
no
A
yes
btwn end-
pts
no
A
yes
57-1



37-1
37-2
37-3
XINT2 same as XINT1
yes
no
YLINE < TPCT
no
yes
Compute prob of error (1.0-ATCT)
35-2
Set value of XLOWER & XUPPER
All points used
no
34-2
yes
37-4
HOLD= 1
yes
no
Store value of XLOWER & XUPPER
Print each prob of error
Compare old and new XLOWER and XUPPER
YLINE= YLINE + INC
Compute and print sys Pe
Set value of XLOWER
HOLD= 2
SCALE= 1.
Set value of XUPPER
35-2
Translate sig set to original location
Set value of XLOWER & XUPPER
Call subroutine


38-1
Compute "areas" 1, 2, 3, & 4
Compute "area" of rectangle and sum
Recompute value of INC
AREA >10^{-3}
yes
INC= .1*INC
no
yes
INC= 0.25*INC
no
AREA >10^{-5}
yes
INC= 0.5*INC
38-2
Prompt for greater resolution
entered
no
yes
Initial increment value halved
34-1
RETURN

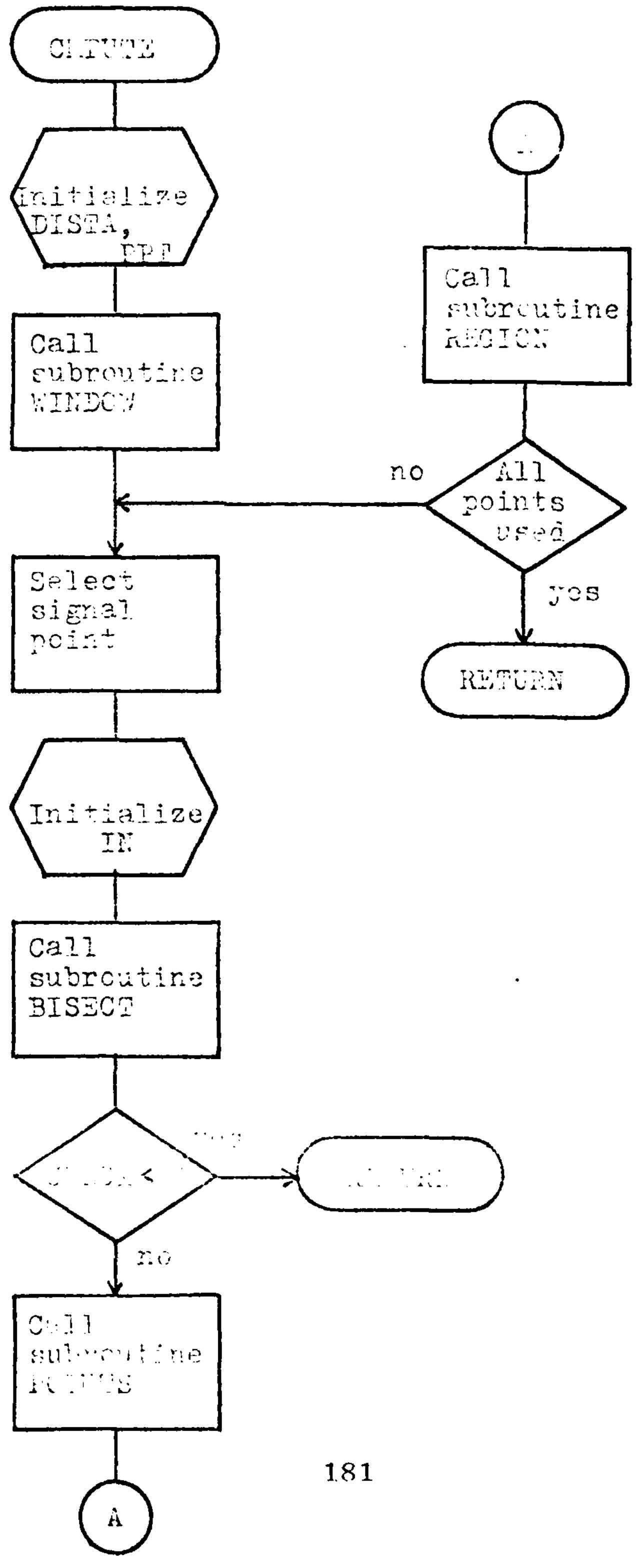

39-1
Determine "XCHECK", x-coord of hort line thru sig pt and the segment
XCHECK > 0
no
yes
Compute value of XLOWER & XUPPER
XUP ≤ XLO
yes
34-2
no
Compute value of XLOWER & XUPPER
XLO ≥ XUP
yes
34-2
no
37-4

AD-A115 498 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOO--ETC F/G 9/2
ANALYSIS OF THE OPTIMUM RECEIVER DESIGN PROBLEM USING INTERACTI--ETC(U)
DEC 81 M R MAZZUCCHI
UNCLASSIFIED AFIT/GE/EE/81D-39 NL

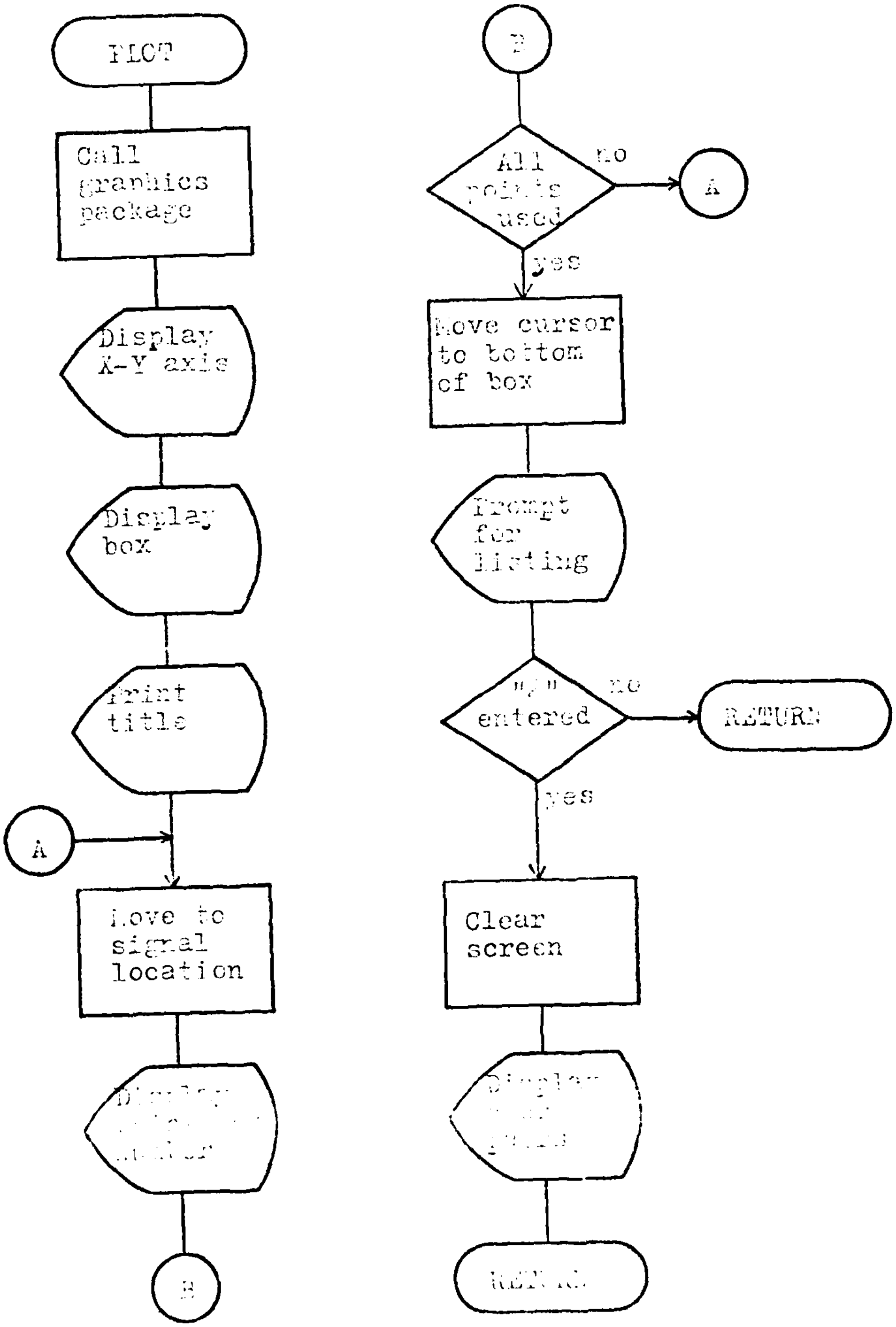
3 OF 3
AD
A115498

END
DATE
FILMED
7 82
DTIC

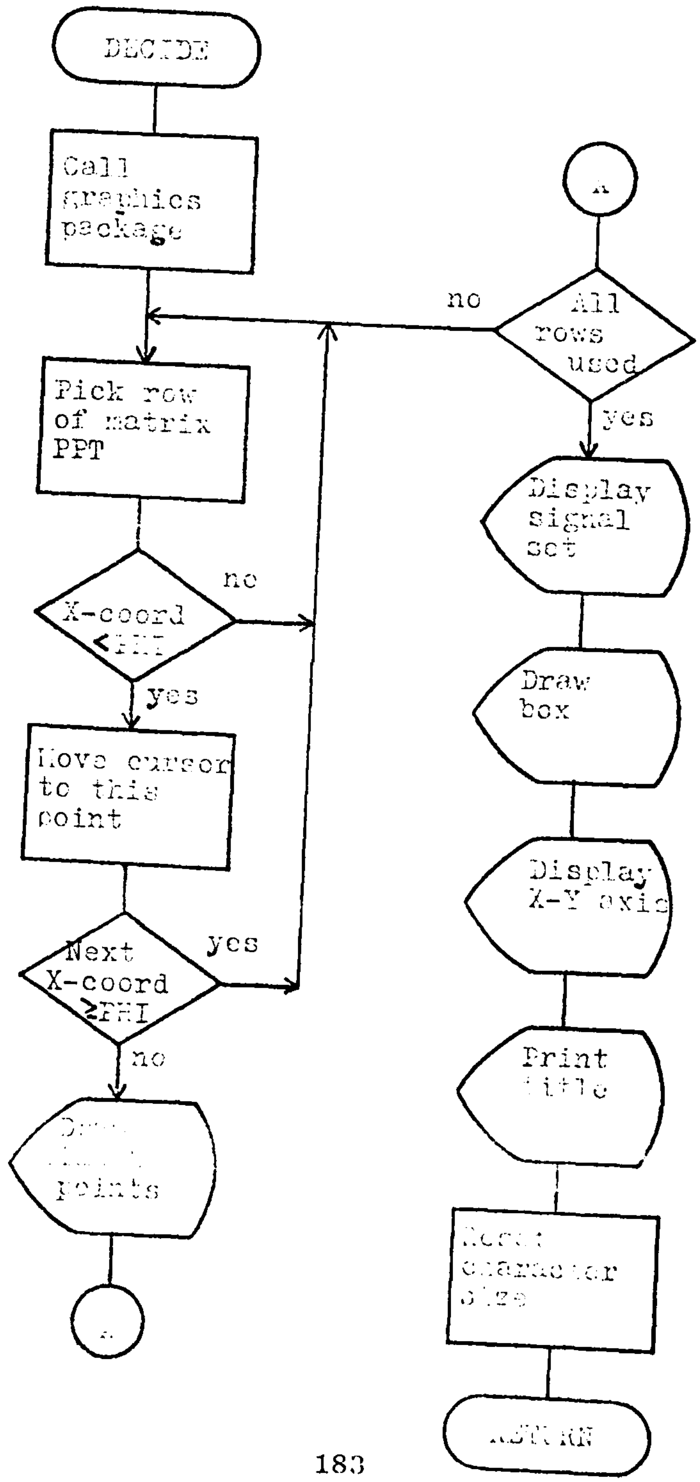
1. Subroutine C[illegible]UTE



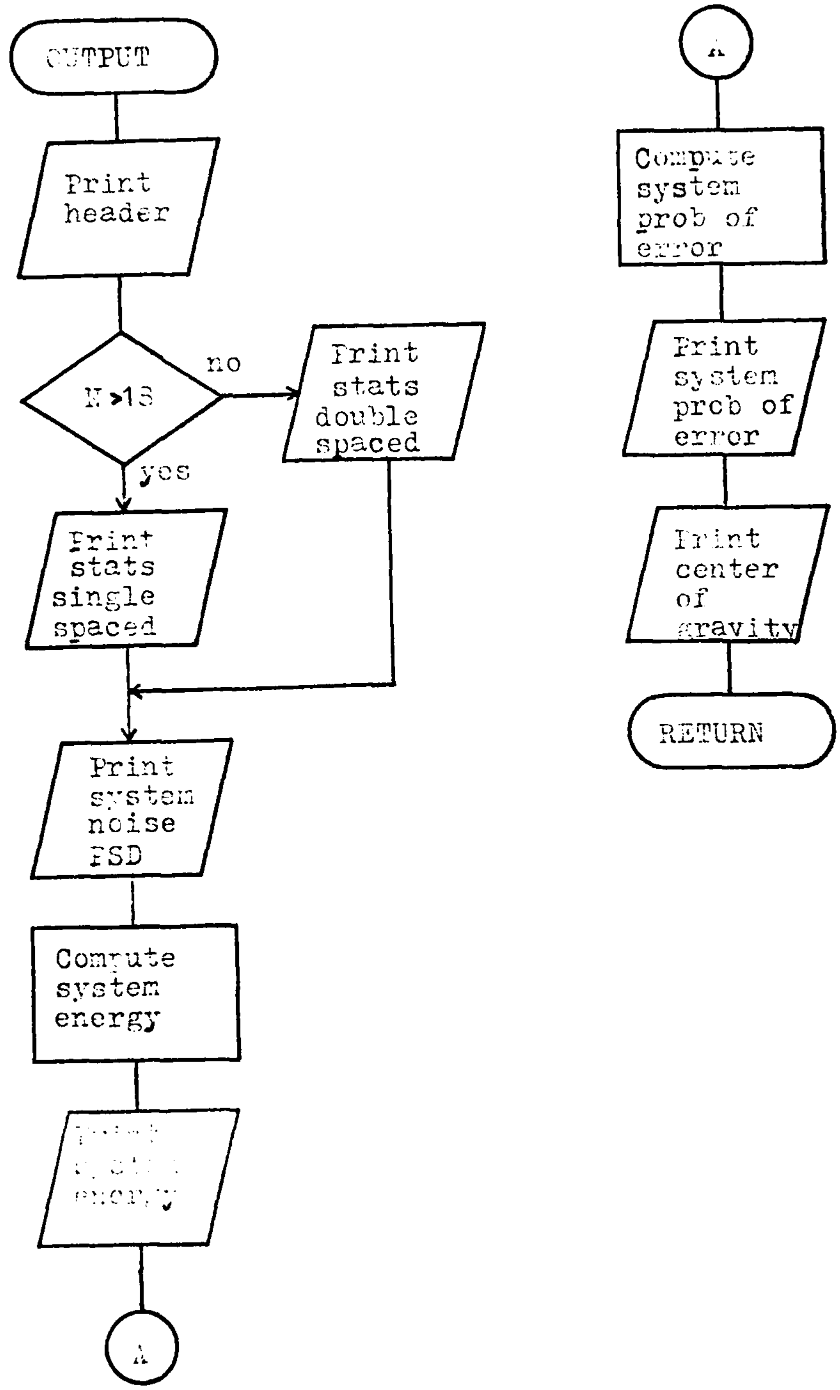
22. Subroutine PLOT



[illegible]. Subroutine DECIDE



24. Subroutine OUTPUT



APPENDIX C

User's Manual for Program SIGDET

Overview of SIGDET

SIGDET is designed as a learning tool for those studying the signal detection and estimation aspects of communications engineering. Specifically, it is an interactive computer program intended for use in the analysis of the optimum communication receiver design problem. Additionally, a graphical capability has been included to provide a pictorial display of the signal set being studied, its decision boundary regions, and system statistics. Some of the features and capabilities include:

* Analysis and manipulation of between 2 and 33 signal points
* Two methods of inputting signal set points
* Ability to add or delete signals any time
* Ability to translate or rotate signal set
* Ability to specify individual signal probabilities and system noise energy
* Provide graphical display of signal set and/or decision regions
* Provide listing of system statistics including probability of error, SNR, center of gravity, signal energies, etc.

The emphasis in the development of the program has been placed on simplicity of use and this manual is intended to demonstrate the range of options available and their

proper use. No attempt is made here to justify the algorithms used or the assumptions inherent in the design of the algorithms.

Initialization and System Flowchart

SIGDET uses the Tektronix PLOT 10 graphics package and in order to use the graphical capability of the program, one of the Tektronix computer terminals must be used. The program is written for use with Tektronix Model 4014 terminal, but will perform adequately on models 4010 and 4012 with little degradation of output display. Additionally, in the computation of the probability of error, several ISML subroutines have been used. Prior to using SIGDET, the graphics package and the ISML routines must be attached for proper execution of the program. This manual provides a sample listing of the CYBER prompts and the user ATTACH and LIBRARY commands necessary.

Once the program has been compiled and is in execution, the user is in control. SIGDET operates in one of two modes. The first is the OPTION mode and upon the prompt *OPTION* > , the user inputs the number of the program function desired. There are currently 14 options available and each is described separately later in this manual. The second mode is the DATA mode. Any time the program requires data, a prompt specifically requesting what information is needed will be given, followed by the > symbol which indicates that SIGDET is waiting for the requested data. If more than

one item per > prompt is required, the inputted data must be separated by a comma. A carriage return by the user signifies all data has been entered and execution is to continue.

The most basic data SIGDET requires are the number of signals in the signal set and a specification of those signals. All other parameters have default values which are assumed unless the user alters them. The flowchart below indicates the main program prompts for inputting this basic data and functions as a road map to assist the user in the operation of the program. One word of caution: once the user specifies an option, he will continue in that option until it has completed its function. In order to terminate the program or start over, he must be in the OPTION mode.

MAIN PROGRAM

A

First Prompt

```
Enter number of signals, 2 to 33
>
```

Second Prompt

```
To enter points with cursor, enter a "2"
To enter points directly, enter a "3"
>
```

Third Prompt ("2" entered)

```
You must provide window dimensions
Enter min and max x-dimensions, please
>
Enter min and max y-dimensions, please
>
```

Screen blanks, window box is drawn, x-y axis indicated by dotted lines, and cursor lines provided.

User positions cross point and enters P to enter point. After all points entered, listing of coordinate pairs is displayed at bottom of screen.

	Enter any digit to continue > is given at end of listings to allow user to copy screen before continuing.
Third Prompt ("3" entered)	*Please enter coordinate pairs.* *Signal 1 >* *Signal 2 >* . . . *Signal n >*
First Notice:	*The default noise energy of 2.0 gives a noise power spectral density or variance of 1.0*
	B
Second Notice:	*The default signal probabilities are equally likely.*
	C
Fourth Prompt	*Option >*

See individual option numbers

Options

The fourteen options currently available in SIGDET are listed below. Those marked with an asterisk are available only on Tektronix Graphics Terminals.

1. Translate signal set

2. Rotate signal set

3. Delete a signal

* 4. Add a signal using graphical cursor

5. Add a signal by specifying coordinate pairs

6. Enter individual signal probabilities

* 7. Display current signal set

* 8. Display decision regions

9. Provide listing of statistics of individual signals and overall system

10. Enter noise energy

11. Scale display boundaries

12. Numerical integration of decision regions

13. Start over

14. Terminate program

OPTION> 1 Translate Signal Set

PURPOSE: To allow user to move entire signal set anywhere in X-Y plane. (Note: Unpredictable results may occur if any signal is allowed to have any energy over 10^6 units.)

PROMPTS: *To translate signal set, enter amount of movement in the X and Y directions*

DELTA X >

DELTA Y >

EXECUTION: The values of DELTA X and DELTA Y are added to the x-y coordinates of each signal and all computations are redone. Hence, the effect is that of maintaining the original frame of reference and relocating all signal points with reference to it.

Point of return to Main Program: C

OPTION> 2 Rotate Signal Set

PURPOSE: To allow user to rotate the signal set about its current X-Y axis.

PROMPTS: *To rotate signal set, enter amount of rotation in degrees, positive or negative, zero to 360.*
ANGLE =

EXECUTION: The input angle is checked to assure that its absolute value is less than or equal to 360. New values for the X-Y coordinates of each point are determined and all computations redone.

Point of return to Main Program: C

OPTION> 3 Delete a Signal

PURPOSE: To allow user to remove unwanted signal without having to start over.

PROMPTS: *Enter the number of the signal to be deleted.*
Delete signal >

EXECUTION: The signal set is reordered with the deleted signal removed. Only one signal may be deleted at a time.

Point of return to Main Program: B

OPTION> 4 Add a Signal Via Cursor

PURPOSE: To allow user to add a signal using the cursor provided with the graphics package.

PROMPTS: Screen will blank and current signal set will be displayed. Under the display, the following prompt is given:

Enter a "2" to clear screen and obtain listing of signal point coordinates. Otherwise enter a "3".

EXECUTION: User must input a "3" at this prompt in order to have cursor available. After positioning the cross point, user types in a "P" to enter

point into signal set. At bottom of screen, coordinates of new point will be provided.

NOTE: Point entered must be within box displayed, otherwise the following error message may appear:

Error encountered, check input and reenter.

Program then returns to A

In order to enter a point outside box via cursor, user should rescale first. (Option 11)

Point of return to Main Program: B

OPTION> 5 Add a Signal Via Specification

PURPOSE: To allow user to add a signal by entering the X and Y coordinates of the new point.

PROMPTS: *Enter X and Y coordinates of added signal*
X coordinate >
Y coordinate >

EXECUTION: The added signal coordinates are included in the signal set and all computations are redone.

Point of return to Main Program: B

OPTION> 6 Enter Signal Probabilities

PURPOSE: To allow the user to specify the probability of each signal.

PROMPTS: *Please enter signal probabilities*
Signal 1 >
Signal 2 >
.
.
.
Signal n >

EXECUTION: The equally likely default signal probabilities are superseded by user entered values. Note: If the sum of the entered probabilities is not in the range $0.95 \geq P \geq 1.05$, the following error message will be given:

Sum of signal probabilities is not within 5% of unity. Please reenter more accurate signal probabilities.

The original prompts are then given again.

Point of return to Main Program: C

<u>OPTION> 7</u> <u>Display Current Signal Set</u>

PURPOSE: To provide user with graphical display of the current signal set and a listing of signal point coordinates.

PROMPTS: Screen will blank and current signal set will be displayed. The X-Y axis is displayed as dotted lines. At bottom of display, the following prompt is given:

Enter a "2" to clear screen and obtain listing of signal point coordinates. Otherwise enter a "3".

EXECUTION: The pause is provided in order to allow the user to copy the display, if desired. If "2" is typed in, screen will clear and coordinates of signals will be displayed accurate to 1/100 of a unit. If a "3" is typed in, return to main program is initiated without clearing the screen.

Point of return to Main Program: C

<u>OPTION> 8</u> <u>Display of Decision Regions</u>

PURPOSE: To provide user with graphical display of the signal set along with the decision boundary regions for each signal.

PROMPTS: No prompts are given.

EXECUTION: Screen will blank and display of signal set with decision boundary regions provided.

Point of return to Main Program: C

OPTION> 9 Signal and System Statistics

PURPOSE: To provide user with the following statistics for each signal:

+ Signal coordinates
+ Probability
+ Energy
+ Signal to Noise Ratio
+ Union Bound on Probability of Error

Provides the following system statistics:

+ System Noise PSD
+ Total System Energy
+ Total Probability of Error
+ System Center of Gravity

PROMPTS: No prompts are given.

EXECUTION: Appropriate headings are provided and tabular listing of indicated statistics typed out.

Point of return to Main Program: C

OPTION> 10 Enter Noise Energy

PURPOSE: To allow user to specify the system noise energy.

PROMPTS: *For a max SNR of zero db, noise energy must be:* xxx
The desired noise energy is >

After user enters noise energy the notice below appears:

This yields a noise PSD or variance of: yyy

EXECUTION: The default noise energy which provides a PSD or variance of 1.0 is superseded by user entered value. The signal having the greatest energy is determined and the noise energy required to result in a SNR of zero db is determined. This value is provided to the user for "ranging" purposes.

Point of return to Main Program: C

OPTION> 11 Enter Scaling Factor

PURPOSE: To allow user the capability of expanding dimensions of displayed signal set. This allows the display of previously "cut off" boundary regions or the adding of signal point via the cursor outside original region.

PROMPTS: *To scale window dimensions, enter scaling factor.*
To reduce window size, factor must be between zero and one.
To increase window size, factor must be greater than one.
Scale factor >

EXECUTION: The original dimensions are multiplied by the scale factor. Note: Unpredictable results may occur when reducing dimensions if scaling causes signals to be outside new dimensions.

Point of return to Main Program: C

OPTION> 12 Numerical Integration of Probability of Error

PURPOSE: To provide user with a very close estimation of the probability of error of each signal and the system.

EXECUTION: Each decision region is approximated by hundreds of rectangular regions. Integration of the Bivariate Normal Density over each rectangle provides the probability of a correct decision. Subtraction of this value from unity yields the probability of error. After the calculation is complete, a tabular listing of each signal, its probability of occurrence, and probability of error is provided.

PROMPTS: At conclusion of the output listing, the user is provided the prompt:

For greater resolution, enter a "2".
Otherwise, enter any other digit.

If a closer estimation is desired, the user responds by entering a "2" and calculation is done again using tighter parameters. Note: The greater the resolution, the greater CPU

time required. In general, the first computation requires about one second of CPU time per signal point.

Point of return to Main Program: C

OPTION> 13 Start Over

PURPOSE: To allow user to return to beginning of program without having to recompile it.

PROMPTS: Since program returns to the beginning, the initial prompts are again provided.

EXECUTION: Returns to start of Main Program.

Point of return to Main Program: A

OPTION> 14 Terminate

PURPOSE: Terminate program.

PROMPTS: No prompts are given.

EXECUTION: Ends program execution.

Vita

Michael Reno Mazzucchi was born 4 December 1949 in Fort Sill, Oklahoma. He graduated from Vicenza American High School in Vicenza, Italy, in 1968 and then attended Purdue University, West Lafayette, Indiana, from which he received the degree Bachelor of Science in Electrical Engineering in June 1972. Upon graduation, he was commissioned as a Second Lieutenant in the United States Army and attended and graduated from the Signal Officer Basic Course, Fort Gordon, Georgia. Subsequently, he served tours with the 85th Maintenance Battalion, 3rd Support Command, Hanau, Germany, and the 10th Transportation Battalion, 7th Transportation Group, Fort Eustis, Virginia, as Signal Officer and Company Commander. From July 1979 to May 1980 he attended the Signal Officer Advanced Course in Fort Gordon and the Telecommunications Staff Officer Course, Kessler Air Force Base, Mississippi. After graduation, he entered the School of Engineering, Air Force Institute of Technology in June 1980. He is married to the former Linda Leigh Crenshaw and they have two children.

Permanent Address: 8917 Galena Drive
El Paso, Texas 79904

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFIT/GE/EE/81D-39	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) ANALYSIS OF THE OPTIMUM RECEIVER DESIGN PROBLEM USING INTERACTIVE COMPUTER GRAPHICS		5. TYPE OF REPORT & PERIOD COVERED MS THESIS December 1981 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Michael R. Mazzucchi Cpt USA		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Institute of Technology (AFIT-EN) Wright-Patterson AFB, Ohio 4533		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS HQDA, MILPERCEN (DAPC-OPP-E), 200 Stovall Street, Alexandria, Va. 22332		12. REPORT DATE December 1981 13. NUMBER OF PAGES 207
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

15 APR 1982

18. SUPPLEMENTARY NOTES

Approved for public release; LAW AFR 190-17

FREDERIC C. LYNCH, Maj, USAF

Director of Public Affairs

Dean for Research and Professional Development

Air Force Institute of Technology (ATC)

Wright-Patterson AFB, OH 45433

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Signal Detection
Computer Aided Instruction
Computer Graphics
Receiver Design

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The purpose of this project was to design a tutorial aid for the study of signal detection theory and its application to communication receiver design. An interactive computer program was developed to solve problems concerning the detection of amplitude and/or phase shift keyed signals in the presence of additive white gaussian niose. The probability of error criterion was used to compare and optimize signal set parameters. (continued)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(continued from block 20)

The user may input from 2 to 33 two-dimensional signal vectors ranging in amplitude from 10^{-6} to 10^{-4} units, specify signal probabilities, and system noise energy. The computed system and signal statistics include signal energy, signal-to-noise ratio, Union Bound on and integrated values of probability of error, noise power spectral density, and center of gravity. Graphical displays provide signal set with coordinates and decision region boundaries. Modification to signal set may be performed via translation, rotation, or scaling, and deletion or addition of signals.

The programming language used was FORTRAN 77 with graphical capability provided thru the Tektronix PLOT-10 graphics package. The program (less graphical capability) may be executed from any interactive terminal supported by the FORTRAN 77 compiler and the International Mathematical & Statistical Libraries (IMSL) routines MDNOR and MDBNOR. For graphical displays, use of Tektronix terminals model 4014, 4012, or 4010 is required.

DATE
ILME
8